

**KLASIFIKASI JENIS BUAH APEL LOKAL BERDASARKAN
PENCIRI WARNA, *ASPECTRATIO* DAN GLCM
MENGUNAKAN *BELT* KONVEYOR BERBASIS RASPBERRY PI**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun oleh:
Lita Nur Fitriani
NIM: 145150301111040



PROGRAM STUDI TEKNIK KOMPUTER
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

LEMBAR PENGESAHAN

KLASIFIKASI JENIS BUAH APEL LOKAL BERDASARKAN PENCIRI WARNA,
ASPECTRATIO DAN GLCM MENGGUNAKAN *BELT* KONVEYOR BERBASIS
RASPERRY PI

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun Oleh :
Lita Nur Fitriani
NIM: 145150301111040

Skripsi ini telah diuji dan dinyatakan lulus pada
19 Oktober 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Dr. Eng. Fitri Utaminingrum, S.T., M.T.
NIP. 19820710 200812 2 001

Dosen Pembimbing II



Wijaya Kurniawan, S.T., M.T.
NIP. 19820125 201504 1 002

Mengetahui

Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T., M.T., Ph.D
NIP. 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Penulis menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 25 Oktober 2018



Lita Nur Fitriani

NIM: 145150301111040

KATA PENGANTAR

Puja dan puji syukur atas kehadiran Allah SWT karena berkat rahmat serta hidayah-Nya, penulis dapat menyelesaikan penelitian untuk memperoleh gelar Sarjana Teknik yang berjudul “Klasifikasi Jenis Buah Apel Lokal Berdasarkan Penciri Warna, *Aspekratio* dan GLCM Menggunakan *Belt* Konveyor Berbasis Raspberry Pi”.

Dalam penyusunan dan penulisan skripsi ini tidak lepas dari bantuan-bantuan yang diberikan oleh berbagai pihak, maka penulis mengucapkan banyak-banyak terima kasih kepada :

1. Bapak Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D. selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya Malang.
2. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D. selaku Ketua Jurusan Teknik Informatika Universitas Brawijaya Malang.
3. Bapak Sabriansyah Rizqika Akbar, S.T., M.Eng. selaku Ketua Program Studi Teknik Komputer Universitas Brawijaya Malang.
4. Ibu Dr. Eng. Fitri Utaminingrum, S.T, M.T selaku dosen pembimbing I yang telah memberikan pengarahan, bimbingan dan semangat sehingga dapat menyelesaikan skripsi ini dengan baik.
5. Bapak Wijaya Kurniawan, S.T, M.T selaku dosen pembimbing II yang telah memberikan pengarahan dan bimbingan kepada peneliti, sehingga dapat menyelesaikan skripsi ini dengan baik.
6. Bapak M. Arifudin dan Ibu Harnanik sebagai orang tua atas segala nasehat, semangat dan dorongan baik berupa do’a maupun materi selama penulis melakukan penelitian.
7. Bapak Abdurrahman Mbawa dan Ibu Artimah sebagai kakek dan nenek serta keluarga besar atas segala nasehat, semangat dan dorongan baik berupa do’a maupun materi selama penulis melakukan penelitian.
8. Bapak Miselan dan Ibu Mujiati sebagai kakek dan nenek serta keluarga besar atas segala nasehat, semangat dan dorongan baik berupa do’a maupun materi selama penulis melakukan penelitian.
9. Teman-teman SMP Rachmilia A, Dewi K.A yang selalu mendukung dan mendoakan dalam mengerjakan skripsi ini.
10. Teman-teman SMK Dindadesi R.A, Noviani S, Karlina Y.A, Fairuzatul M yang selalu mendengar keluh kesah penulis serta mendukung dan mendoakan dalam mengerjakan skripsi ini.
11. Burhan M, Linda S.P, Muzammilatul J, Ida Y, Intan F, Putri A.D, Yongki P, Oggy S, Adrian M.P, Syahriel D dan kontrakan ceria yang selalu membantu dan memberikan semangat serta motivasi selama proses pengerjaan skripsi ini.
12. Seluruh teman-teman Teknik Komputer 2014 yang selalu mendukung dan memotivasi penulis dalam mengerjakn skripsi ini.
13. Seluruh pihak yang tidak dapat disebutkan satu persatu yang telah berperan dalam penyelesaian skripsi ini.

Penulis menyadari dalam penyusunan skripsi ini terdapat banyak kekurangan, sehingga saran dan kritik yang membangun sangat dibutuhkan untuk ke depannya agar lebih baik lagi. Semoga penelitian ini dapat memberikan manfaat bagi semua pihak yang membacanya.

Malang, 25 Oktober 2018

Lita Nur Fitriani

lita.nur.fitriani@gmail.com



ABSTRAK

Kota Batu terletak pada ketinggian rata-rata 871 mdpl. Sehingga membuat Kota Batu memiliki berbagai macam hasil bumi. Salah satu hasil perkebunan yang berlimpah adalah Buah apel. Buah dengan nama latin *Malus sylvestris* Mill merupakan komoditas utama hasil perkebunan. Terdapat empat jenis apel yaitu Apel Anna, Manalagi, Wanglin dan Rome Beauty. Dari keempat jenis apel tersebut pada saat dilakukan panen lalu dilakukan sortir berdasarkan jenisnya, proses ini masih menggunakan tenaga manusia. Tentunya proses ini sering kali tidak akurat karena proses pemilihan yang dilakukan dapat berbeda-beda tiap orang, karena bedanya persepsi tiap orang. Berdasarkan permasalahan tersebut dibuat sebuah sistem sortir yaitu dengan memanfaatkan klasifikasi yang dapat memisahkan keempat jenis apel berdasarkan bentuk, warna, dan tekstur. Dalam sistem ini menggunakan *Webcam* sebagai sensor untuk menangkap gambar buah apel lalu di proses pada Raspberry Pi 3. Untuk melakukan proses pemilahan dengan menggunakan tiga buah servo sebagai aktuator untuk mendorong buah apel masuk pada klasifikasinya. Citra yang sudah ditangkap oleh *webcam* selanjutnya akan diproses pada Raspberry Pi lalu gambar tersebut akan dilakukan metode pengolahan citra sehingga mendapatkan nilai *Hue*, *Aspectratio* dan *GLCM Contrast*. Jika nilai sudah didapatkan maka Raspberry Pi 3 dan Arduino Uno berkomunikasi menggunakan komunikasi serial I2C sehingga servo akan bergerak sesuai hasil klasifikasi. Dari percobaan yang telah dilakukan didapatkan hasil akurasi nilai *aspectratio* sebesar 80%. Untuk pengujian akurasi antara *software* dan *hardware* sebesar 80%. Sedangkan rata-rata waktu komputasi sebesar 15997.2 ms atau sebesar 15 detik.

Kata Kunci : *GLCM, Konveyor, Buah Apel Batu, Sortir.*

ABSTRACT

*Batu City is located at an average altitude of 871 m below sea level. Thus, it makes Batu City has a variety kind of produce. One of the abundant product of plantations is apples. Fruit with the Latin name *Malus sylvestris* Mill is the main commodity of the product of plantation. There are four types of apples, such as Anna Apple, Manalagi Apple, Wanglin Apple, and Rome Beauty Apple. From the four types of apples, when crop is done, and then sorting will be done based on its type, this process still uses human power. Certainly, this process is often inaccurate because the process of selection done can be different for each person, because of the different perceptions of each person. Based on these problems, a sorting system is made by utilizing a classification that can separate the four types of apples based on shape, color, and texture. In this system, it uses a Webcam as a sensor to capture images of apples, and then it is processed on Raspberry Pi 3. The process of sorting uses three servos as an actuator to push the apples into its classification. The image that has been captured by the webcam will be processed on Raspberry Pi, and then the image will be done with image processing method to get the Hue, Aspectratio and GLCM Contrast values. If the value has been obtained, Raspberry Pi 3 and Arduino Uno communicate by using I2C serial communication, so that the servo will move based on the result of classification. From the study that has been done, it is obtained the result of the accuracy of aspectratio value as much as 80%. For testing, the accuracy between software and hardware is as much as 80%. While the average time of computation is as much as 15997.2 ms or 15 seconds.*

Keywords : GLCM, Conveyor, Batu Apple, Sorting.

DAFTAR ISI

LEMBAR PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	2
1.5 Batasan masalah	3
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Tinjauan Pustaka	5
2.2 Dasar Teori	6
2.2.1 Buah Apel Lokal	6
2.2.2 <i>Computer Vision</i>	8
2.2.3 Citra Digital	8
2.2.4 Tekstur	11
2.2.5 Ukuran Citra Digital	13
2.2.6 <i>Bounding Box</i>	14
2.2.7 <i>Aspectratio</i>	14
2.2.8 <i>Morphology</i>	15
2.2.9 <i>Thresholding</i>	16
2.2.10 Webcam Logitech	17
2.2.11 Raspberry Pi 3.....	17
2.2.12 Motor Servo.....	18
2.2.13 Motor Stepper	19

2.2.14 Driver Motor A4988.....	20
BAB 3 METODOLOGI	21
3.1 Studi Literatur	21
3.2 Analisis Kebutuhan	22
3.3.1 Kebutuhan Perangkat Keras.....	22
3.3.2 Kebutuhan Perangkat Lunak	22
3.3 Perancangan Sistem.....	22
3.3.1 Perancangan Perangkat Keras.....	23
3.3.2 Perancangan Perangkat Lunak	23
3.4 Implementasi	23
3.5 Pengujian dan Analisis	24
3.6 Kesimpulan.....	24
BAB 4 REKAYASA KEBUTUHAN.....	25
4.1 Gambaran Umum Sistem.....	25
4.1.1 Tujuan.....	25
4.1.2 Karakteristik Pengguna	25
4.1.3 Lingkungan Operasi	25
4.1.4 Batasan Perancangan dan Implementasi	26
4.1.5 Asumsi dan Ketergantungan	26
4.2 Analisis Kebutuhan Sistem	26
4.2.1 Kebutuhan Fungsional.....	26
4.2.2 Kebutuhan Non Fungsional.....	27
4.3 Pengumpulan Data	29
BAB 5 PERANCANGAN DAN IMPLEMENTASI	30
5.1 Perancangan Sistem.....	30
5.1.1 Perancangan <i>Prototype</i> Klasifikasi Buah Apel	30
5.1.2 Perancangan Perangkat Keras.....	31
5.1.3 Perancangan Perangkat Lunak	33
5.2 Implementasi Sistem	41
5.2.1 Implementasi <i>Prototype</i> Alat Klasifikasi Jenis Apel	41
5.2.2 Implementasi Perangkat Keras.....	42
5.2.3 Implementasi Perangkat Lunak	45

BAB 6 PENGUJIAN DAN ANALISIS.....	59
6.1 Pengujian Akurasi Nilai <i>Aspctratio</i>	59
6.1.1 Tujuan Pengujian	59
6.1.2 Prosedur Pengujian	59
6.1.3 Pelaksanaan Pengujian	59
6.1.4 Hasil dan Analisis Pengujian	59
6.2 Pengujian Waktu Komputasi.....	62
6.2.1 Tujuan Pengujian	62
6.2.2 Prosedur Pengujian	63
6.2.3 Pelaksanaan Pengujian	63
6.2.4 Hasil dan Analisis Pengujian	63
6.3 Pengujian Akurasi Klasifikasi Dari Sisi <i>Software</i> Dan <i>Hardware</i>	65
6.3.1 Tujuan Pengujian	65
6.3.2 Prosedur Pengujian	65
6.3.3 Pelaksanaan Pengujian	66
6.3.4 Hasil dan Analisis Pengujian	66
BAB 7 PENUTUP	70
7.1 Kesimpulan.....	70
7.2 Saran	70
DAFTAR PUSTAKA.....	71
LAMPIRAN A HASIL PENGUJIAN	74
LAMPIRAN B KODE PROGRAM PENGOLAHAN CITRA	76
LAMPIRAN C KODE PROGRAM ARDUINO	82
LAMPIRAN D NILAI DATA UJI PENGAMBILAN <i>RANGE</i>	85

DAFTAR TABEL

Tabel 2.1 Fitur GLCM.....	12
Tabel 2.2 Spesifikasi Raspberry Pi	17
Tabel 2.3 Spesifikasi Motor Servo MG995	18
Tabel 2.4 Spesifikasi Motor Stepper Nema 17	19
Tabel 5.1 Pin antara <i>Motor Driver</i> dengan Power Supply	31
Tabel 5.2 Pin antara Motor <i>Stepper</i> dengan Motor <i>Driver</i>	32
Tabel 5.3 Pin antara Motor <i>Driver</i> dengan Arduino Uno	32
Tabel 5.4 Pin antara GPIO Raspberry Pi 3 dengan Arduino Uno	32
Tabel 5.5 Koneksi antara Motor Servo dengan Power Supply	33
Tabel 5.6 Pin antara Motor Servo dengan Arduino Uno	33
Tabel 5.7 Menyambungkan Webcam dengan Raspberry Pi 3	33
Tabel 5.8 Inisialisasi Library Pada Raspberry Pi 3.....	45
Tabel 5.9 Kode Program Motor Stepper	46
Tabel 5.10 Inisialisasi Webcam	46
Tabel 5.11 Membaca Nilai Webcam	47
Tabel 5.12 Pengolahan Citra Digital.....	47
Tabel 5.13 Pendeteksian Objek.....	47
Tabel 5.14 Proses Klasifikasi.....	55
Tabel 5.15 Kode Program Motor Servo	56
Tabel 6.1 Pengujian Nilai Aspectratio Apel Ana.....	59
Tabel 6.2 Pengujian Aspectratio Apel Manalagi	60
Tabel 6.3 Pengujian Aspectratio Apel Romebeauty	61
Tabel 6.4 Pengujian Aspectratio Apel Wanglin.....	61
Tabel 6.5 Rata-rata akurasi <i>aspectratio</i>	62
Tabel 6.6 Pengujian Waktu Komputasi Apel Ana.....	63
Tabel 6.7 Pengujian Waktu Komputasi Apel Manalagi	64
Tabel 6.8 Pengujian Waktu Komputasi Apel Romebeauty	64
Tabel 6.9 Pengujian Waktu Komputasi Apel Wanglin.....	64
Tabel 6.10 Hasil rata-rata waktu komputasi keempat apel	65
Tabel 6.11 Akurasi <i>Hardware</i> dan <i>Software</i> saat mendorong Apel Ana	66
Tabel 6.12 Akurasi <i>Hardware</i> dan <i>Software</i> saat mendorong Apel Manalagi.....	67
Tabel 6.13 Akurasi <i>Hardware</i> dan <i>Software</i> saat mendorong Apel Romebeauty	67
Tabel 6.14 Akurasi <i>Hardware</i> dan <i>Software</i> saat mendorong Apel Manalagi.....	68
Tabel 6.15 Rata-rata akurasi kesesuaian <i>Hardware</i> dan <i>Software</i>	68

DAFTAR GAMBAR

Gambar 2.1 Apel Anna	7
Gambar 2.2 Apel Manalagi.....	7
Gambar 2.3 Apel Wanglin	7
Gambar 2.4 Apel Rome Beauty	8
Gambar 2.5 Gambar hasil konversi RGB ke Grayscale	9
Gambar 2.6 Hasil konversi dari RGB ke HSV	10
Gambar 2.7 Hasil konversi dari RGB ke YCbCr	11
Gambar 2.8 Ruang citra kontinyu dan gambar digital	13
Gambar 2.9 Contoh citra dalam piksel yang berbeda	14
Gambar 2.10 Gambar yang sudah di <i>bounding box</i>	14
Gambar 2.11 Hasil operasi dilasi.....	15
Gambar 2.12 Hasil operasi erosi	15
Gambar 2.13 Hasil operasi <i>opening</i>	16
Gambar 2.14 Hasil operasi <i>closing</i>	16
Gambar 2.15 Segmentasi menggunakan <i>thresholding</i>	16
Gambar 2.16 Webcam Logitech.....	17
Gambar 2.17 Raspberry Pi 3.....	18
Gambar 2.18 Motor Servo MG995	19
Gambar 2.19 Stepper Motor.....	20
Gambar 2.20 IC A4988	20
Gambar 3.1 Diagram Alir Metodologi Penelitian.....	21
Gambar 3.2 Diagram Blok Peancangan Perangkat Keras	23
Gambar 3.3 Diagram Blok Perancangan Perangkat Lunak	23
Gambar 5.1 Desain Prototype Alat Klasifikasi Buah Apel	30
Gambar 5.2 Skematik Rangkaian Perangkat Keras	31
Gambar 5.3 Flowchart Program Utama Sistem	34
Gambar 5.4 Flowchart <i>Stepper</i>	35
Gambar 5.5 Flowchart Pengambilan Data Citra	36
Gambar 5.6 Flowchart Pengolahan Citra Apel.....	37
Gambar 5.7 Flowchart Pengambilan Data Uji.....	38
Gambar 5.8 Flowchart Proses GLCM	39
Gambar 5.9 Flowchart Klasifikasi	40
Gambar 5.10 Flowchart Menjalankan Servo	41
Gambar 5.11 Tampilan Alat Secara Keseluruhan.....	42
Gambar 5.12 Menghubungkan Motor Stepper dengan Driver	42
Gambar 5.13 Menghubungkan Driver dengan Arduino Uno.....	43
Gambar 5.14 Menghubungkan Raspberry Pi 3 dengan Webcam Logitech.....	43
Gambar 5.15 Tampilan Webcam Logitech.....	44

Gambar 5.16 Menghubungkan Raspberry Pi 3 dengan Arduino Uno	44
Gambar 5.17 Menghubungkan Arduino Uno dengan Motor Servo	44
Gambar 5.18 (a) Citra yang sudah di bounding box, (b) Citra dikonversi ke YCbCr, (c) Hasil Opening, (d) Hasil opening digabungkan dengan citra asli.....	50
Gambar 5.19 Pengukuran manual dengan menggunakan penggaris.....	51
Gambar 5.20 Gambar buah apel yang ditangkap oleh webcam	52
Gambar 5.21 Nilai lebar dan tinggi	52
Gambar 5.22 Pengukuran manual dengan menggunakan penggaris.....	53
Gambar 5.23 Gambar buah apel yang ditangkap oleh webcam	53
Gambar 5.24 Nilai lebar dan tinggi	53
Gambar 5.25 Nilai RGB.....	52
Gambar 6.1 Hasil Pengujian <i>Aspectratio</i>	62
Gambar 6.3 <i>Output</i> Waktu Komputasi	65



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Kota Batu merupakan salah satu kota yang baru terbentuk pada tahun 2001 sebagai pecahan dari Kabupaten Malang, Jawa Timur. Kota Batu terletak pada ketinggian 680-1900 diatas permukaan laut. Sehingga membuat Kota Batu memiliki berbagai macam hasil bumi. Salah satu hasil perkebunan yang berlimpah adalah Buah apel. Buah dengan nama latin *Malus sylvestris* Mill merupakan komoditas utama hasil perkebunan (Yuwono, 2015).

Buah Apel merupakan produk khas yang menjadi andalan di Kota Batu, terdapat 4 (empat) jenis yang memiliki rasa, bentuk dan tekstur kulit yang berbeda. Keempat buah apel tersebut adalah apel Ana, Manalagi, Romebeauty, dan Wanglin. Keempat jenis buah tersebut tumbuh subur di Kota Batu sehingga hasil buah apel menjadi berlimpah. Berlimpahnya hasil panen buah Apel tentunya membuat petani gembira karena tentunya dapat meningkatkan omset petani. Berlimpahnya hasil panen tersebut juga diiringi oleh permintaan pasar yang banyak.

Berlimpahnya hasil panen akan berdampak pada proses sortir yang memilah berdasarkan jenisnya. Proses pemilahan ini masih dilakukan secara manual sehingga membutuhkan banyak tenaga manusia untuk melakukan pemilahan. Proses ini sering kali tidak akurat karena proses pemilihan yang dilakukan dapat berbeda-beda tiap orang, karena bedanya persepsi tiap orang, sehingga sangat memungkinkan terjadi kesalahan. Apabila klasifikasi dilakukan bagi orang yang tidak pakar, maka sangat memungkinkan terjadi kesalahan karena tidak semua orang bisa membedakan atau mengklasifikasi jenis apel secara virtual tanpa adanya pengalaman.

Adapun penelitian sebelumnya yang telah dilakukan oleh Al Amin Putra dengan judul “Perancangan Dan Pembuatan Alat Pemisah Buah Apel Berdasarkan Ukuran Dengan Pengendali Mikrokontroler Atmega 8535” (AL Amin Putra, 2013). Pemisahan buah apel berdasarkan ukuran secara otomatis dengan menggunakan sensor laser dan photodiode sebagai pendeteksi ukuran buah, dan hasil bacaan dari sensor akan diampilkkan pada LCD. Setelah mendapat hasil sensor kan mengirim data ke Mikrokontroller Atmega 8535 untuk mengaktifkan motor DC guna memasukkan apel ke wadah berdasarkan ukuran.

Penelitian yang berjudul “Perancangan Sistem Inspeksi Visual Berbasis Computer Vision Untuk Penggolongan Buah Apel” yang ditulis oleh Syakir Almas Amrullah (Amrullah, 2017) menerapkan sistem *computer vision* untuk mengekstraksi dan menganalisis penciri warna dan bentuk pada citra buah apel. Sensor yang digunakan untuk menangkap citra kulit buah apel dengan menggunakan Webcam Logitech C170 dan dibantu dengan lampu LED strip sedangkan perangkat lunak yang digunakan adalah *Visual Studio Community 2015* dan pustaka *OpenCV 3.0.0*. Proses program yang dilakukan untuk mendeteksi yaitu dilakukan proses akuisisi citra, deteksi tepi dengan menggunakan *Canny* lalu

dilakukan ekstraksi penciri bentuk dan ekstraksi penciri warna. Tahapan yang dilakukan dalam ekstraksi penciri warna dengan mentransformasikan citra ruang warna RGB ke dalam ruang warna HSV. Ruang warna HSV lebih menyerupai dan sesuai dengan persepsi manusia. Dalam penelitian ini hanya menggunakan kanal warna *Hue*, ini dikarenakan lebih baik meskipun mendapat pencahayaan yang berbeda. Pada penelitian ini hanya menggunakan buah apel anna, fuji dan *red delicious*.

Berdasarkan permasalahan yang diangkat, diperlukan sebuah solusi yang efisien untuk melakukan proses sortir dengan cara yang tepat. Dibangunlah sebuah sistem sortir dengan memanfaatkan konveyor yang dapat memilah berdasarkan warna, bentuk dan tekstur. Menggunakan *webcam* sebagai sensor untuk mendeteksi suatu objek yang akan diteliti. Dengan menaruh buah apel pada sabuk konveyor lalu citra objek diolah pada sebuah *software* yaitu *opencv* dengan parameter bentuk, warna kulit dan tekstur kulit. Dari hasil parameter tersebut lalu buah apel akan di klasifikasikan berdasarkan jenisnya dengan menggunakan aktuator berupa lengan pendorong yang digerakkan oleh Motor Servo.

Dalam penelitian ini menggunakan parameter berupa nilai *Hue* dari format warna HSV dan *aspectratio* dimana terdapat nilai tinggi (*Height*) dan lebar (*Width*). Sedangkan untuk mendapatkan hasil tekstur kulit Apel menggunakan metode *Grey Level Co-Occurrence Matrix* (GLCM) dimana GLCM ini bekerja dengan melihat piksel pertama dan tiga piksel di sebelah kiri, kanan, atas bawah, proses ini akan terus diulang hingga piksel terakhir gambar tersebut.

Ketiga parameter tersebut nantinya akan dijadikan inputan untuk selanjutnya dilakukan proses klasifikasi. Proses klasifikasi dengan menggunakan percabangan *if*. Jika hasil sesuai, maka lengan pendorong akan bergerak sesuai jenis klasifikasinya.

1.2 Rumusan masalah

Rumusan masalah dalam penelitian ini yaitu :

1. Bagaimana akurasi nilai *aspectratio* untuk mendeteksi buah apel?
2. Bagaimana waktu komputasi?
3. Bagaimana akurasi klasifikasi dari sisi *software* dan *hardware*?

1.3 Tujuan

Tujuan yang ingin dicapai dari penelitian ini adalah :

1. Dapat menunjukkan akurasi nilai *aspectratio* untuk mendeteksi buah apel
2. Dapat menunjukkan waktu komputasi
3. Dapat menunjukkan klasifikasi buah apel dari sisi *software* dan *hardware*

1.4 Manfaat

Manfaat yang didapat dari penelitian ini adalah :

1. Dapat digunakan sebagai informasi mengenai keempat jenis buah Apel lokal dengan proses yang sudah modern.
2. Dapat diterapkan pada industri sortir buah apel yang tentunya dengan prosesor yang besar.
3. Dapat meringankan dan memudahkan pekerjaan manusia.

1.5 Batasan masalah

Adapun Batasan masalah pada sistem yang dibuat adalah sebagai berikut :

1. Menggunakan 4 (empat) jenis buah Apel lokal yaitu Manalagi, Anna, Wanglin dan Rome Beauty.
2. Proses pengambilan gambar dilakukan di dalam ruangan (*indoor*).
3. *Background* dan sabuk (*belt*) dari *conveyor* berwarna biru polos.
4. Proses pengolahan citra menggunakan perangkat lunak *opencv* python 3.5
5. Posisi kamera tetap yaitu berada di samping salah satu sisi dari *conveyor* yang sudah ditentukan oleh penulis.
6. Sistem hanya menggunakan satu buah *webcam* memiliki resolusi yang besar yaitu 720p untuk merekam video sedangkan untuk memotret memiliki resolusi sebesar 5MP.
7. Implementasi menggunakan mikrokontroller Raspberry Pi 3.
8. Fitur yang digunakan pada GLCM hanya *Contrast*

1.6 Sistematika pembahasan

Uraian singkat mengenai struktur penulisan pada masing-masing bab adalah sebagai berikut :

BAB I PENDAHULUAN

Pada bab ini berisi mengenai latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah dan sistematika pembahasan dari “KLASIFIKASI JENIS BUAH APEL LOKAL BERDASARKAN PENCIRI WARNA, *ASPECTRATIO* DAN GLCM MENGGUNAKAN *BELT* KONVEYOR BERBASIS RASPBERRY PI”

BAB II Landasan Kepustakaan

Pada bab ini menjelaskan tentang landasan teori yang terkait dengan penelitian, menjelaskan tentang penelitian serupa yang pernah dilakukan.

BAB III Metode Penelitian

Pada bab ini membahas tentang alur kerja yang dilakukan dalam penulisan diantaranya studi literatur, analisi kebutuhan, perancangan, implementasi dan pengujian sistem.

BAB IV Rekayasa Persyaratan

Pada bab ini membahas tentang kebutuhan system, kebutuhan fungsional dan non fungsional dari sistem.

BAB V Perancangan dan Implementasi

Menjelaskan tentang perancangan sistem sesuai dengan diagram blok system serta hasil implementasi perancangan sistem.

BAB VI Pengujian

Pada bab ini dilakukan pengujian apabila sistem telah dilakkan perancangan dan implementasi. Pengujian dilakukan untuk melakukan uji coba sistem serta melakukan analisis.

BAB VII Penutup

Pada bab in bersisi tentang saran dan kesimpulan dari sistem yang telah dibuat.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Tinjauan Pustaka

Tinjauan pustaka terdiri dari beberapa jurnal yang dijadikan referensi oleh penulis untuk mendukung penelitian. Beberapa penelitian sebelumnya yang terdapat dalam jurnal atau paper ini, penulis dapat mengerti tentang perkembangan penelitian yang serupa serta dapat ditemukan hasil penelitian yang relevan sehingga mendukung penelitian ini.

Penelitian yang dilakukan oleh (Neneng, 2017) dengan judul “Klasifikasi Jenis Daging Berdasarkan Analisis Citra Tekstur Gray Level Co-Occurrence Matrices (GLCM) dan Warna” dengan menggunakan pengolahan citra digital. Pada penelitian ini klasifikasi jenis citra daging dengan menggunakan tekstur statistik dan warna dengan metode pelatihan dan pengujian menggunakan *Support Vector Machine* (SVM). Untuk mengenali tekstur dari jenis daging dengan menggunakan GLCM. Ciri statistic disini menggunakan adalah *Angular Second Moment* (ASM), *Invers Different Moment* (IDM), entropi, *contrast* dan korelasi. Sedangkan format warna yang digunakan yaitu HSV. Citra daging yang diperoleh dari kamera digital berformat RGB, dan dikonversi ke format HSV. Penggunaan format HSV karena ruang warna yang merepresentasikan warna seperti yang dilihat mata manusia. Ekstraksi ciri tekstur yang digunakan adalah GLCM.

Penelitian yang dilakukan oleh (Febyan Dimas Pramanta, 2017) dengan judul “Sistem Cerdas Penyortir Apel Berdasarkan Warna Dan Ukuran Berbasis Mikrokontroler Arduino” memanfaatkan konveyor dengan *belt* sebagai penggerak buah apel. Pada penelitian ini mendeteksi tiga jenis apel berdasarkan ukuran dan warna dengan menggunakan sensor TCS3200 untuk mendeteksi warna dan sensor LDR untuk mendeteksi ukuran dengan menempatkan kedua sensor tersebut pada pintu masuk buah apel. Kedua parameter tersebut diproses pada mikrokontroler Arduino uno. Dengan menggunakan 2 motor DC 12V sebagai penggerak.

Pada penelitian yang berjudul “Perancangan Dan Implementasi Alat Penyortir Barang Pada Konveyor Dengan Pengolahan Citra” yang dilakukan oleh (Fajar Ridho Wicaksono, 2018) membuat sebuah sistem sortir untuk barang dengan warna merah, biru dan warna lain. Terdapat 3 buah lengan pendorong untuk 3 jenis warna benda yang dideteksi. Benda yang akan dideteksi diletakkan pada sabuk konveyor yang digerakkan oleh Motor DC 24V. Pada penelitian ini menggunakan webcam untuk mendeteksi objek dan menggunakan mikrokontroler yaitu Raspberry Pi 3. Sedangkan untuk mendorong benda memasuki jalur distribusinya dengan menggunakan MotorDC 6V.

2.2 Dasar Teori

2.2.1 Buah Apel Lokal

Buah Apel yang mempunyai nama latin *Malus sylvestris* Mill merupakan tanaman buah tahunan yang tumbuh didaerah dengan iklim sub tropis. Tanaman apel berasal dari daerah Asia Barat dan di Indonesia sendiri apel telah ditanam sejak tahun 1934 hingga saat ini. Di Indonesia, apel dapat tumbuh dan berbuah di daerah dataran tinggi Kota Batu dan Kota Malang yaitu di daerah Poncokusumo dan Nongkojajar merupakan sentra produksi apel. Hal ini dikarenakan Kota Batu terletak pada ketinggian + 680-1.900 mdpl dan 15 km sebelah barat kota Malang, (Yuwono, 2015).

Buah Apel yang berada di Kota Batu memiliki empat jenis. Keempat jenis tersebut adalah Apel Anna, Manalagi, Wanglin dan Rome Beauty. Keempat buah apel tersebut memiliki kandungan vitamin yang terdiri dari vitamin A, B1, B2, B3, B5, B6, B9 dan vitamin C. Kulit Apel juga dapat dikonsumsi karena mengandung senyawa bernama *Triterpenoid* yang bersama-sama dengan *Flavonoid* mencegah pertumbuhan tiap sel kanker (Yosefina, 2015).

Dengan banyaknya kandungan vitamin dan senyawa yang baik untuk tubuh, dapat dikatakan bahwa buah apel merupakan buah yang memiliki daya jual yang tinggi. Apel Anna dan Manalagi merupakan dua jenis apel yang sering ditemui di pasaran dan memiliki harga yang relative lebih murah dibanding dengan dua jenis lainnya. Keempat jenis tersebut memiliki ciri warna, bentuk dan tekstur kulit yang berbeda-beda, sehingga tidak sedikit orang yang merasa kesusahan dalam membedakan. Tetapi, jika diteliti lagi terdapat perbedaan dari keempatnya. Apel Manalagi memiliki bentuk bulat sempurna dengan kulit buah berwarna hijau. Apel Wanglin memiliki bentuk yang bulat juga tetapi terdapat titik-titik atau tekstur kuning yang mengelilingi kulit buah dan warna kulit hijau. Apel Rome Beauty memiliki bentuk bulat sempurna tetapi terdapat corak merah di sekeliling kulit buah. Sedangkan Apel Anna memiliki bentuk lonjong, berwarna merah dan terdapat corak hijau sedikit di kulit buahnya. Keempat jenis tadi adalah :

2.2.1.1 Apel Anna

Apel Anna memiliki bentuk lonjong, berwarna merah dan terdapat corak hijau sedikit di kulit buahnya. Corak merah di sekitar buah apel dikarenakan terpapar sinar matahari. Pada dasarnya yang membedakan apel ana dan ketiga apel yaitu bentuknya yang lonjong. (Sutopo, 2015)



Gambar 2.1 Apel Anna
(Yuwono, 2015)

2.2.1.2 Apel Manalagi

Apel Manalagi memiliki bentuk bulat sempurna dengan kulit buah berwarna hijau. Tidak menutup kemungkinan kulit buah terdapat corak warna merah yang dikarenakan terpapar sinar matahari. (Sutopo, 2015)



Gambar 2.2 Apel Manalagi
Sumber : (Monda, 2015)

2.2.1.3 Apel Wanglin

Apel Wanglin memiliki bentuk yang bulat juga tetapi terdapat titik-titik atau tekstur kuning yang mengelilingi kulit buah dan warna kulit hijau. Buah apel ini mulai sulit ditemui dikarenakan jumlah buahnya sedikit dan hanya beberapa petani saja yang membudidayakan (Hebert Adrianto, 2013).



Gambar 2.3 Apel Wanglin
Sumber : (Auliia, 2013)

2.2.1.4 Apel Rome Beauty

Apel Rome Beauty memiliki bentuk bulat sempurna hingga lonjong. Terdapat corak merah di sekeliling kulit buah. Warna merah ini hanya pada bagian yang terkena sinar matahari, sedangkan bagian yang lain tetap berwarna hijau. (Yuwono, 2015)



Gambar 2.4 Apel Rome Beauty

Sumber : (Wisata, 2015)

2.2.2 Computer Vision

Computer vision dapat dikatakan mencoba meniru cara kerja visual manusia. Visual manusia ini sesungguhnya sangat kompleks. Manusia melihat suatu objek dengan menggunakan mata lalu diteruskan ke otak untuk diinterpretasikan sehingga manusia tersebut mengetahui objek apa. Selayaknya mata dan otak, *computer vision* adalah dapat dikatakan suatu sistem yang mempunyai suatu kemampuan untuk menganalisis objek secara visual, lalu hasil dari objek tersebut berbentuk citra digital (RD. Kusumanto, 2012).

2.2.3 Citra Digital

Citra digital adalah representative dari citra dengan bentuk pendekatan berdasarkan sampling dan kuantisasi yang diambil dari mesin. Sampling pada citra menyatakan besar kecilnya piksel pada citra dan kuantisasi menyatakan besarnya nilai tingkat kecerahan yang dinyatakan dalam nilai tingkat keabuan sesuai dengan jumlah bit biner yang digunakan oleh mesin. Adapaun beberapa jenis citra yang sering digunakan adalah : (Riadi, 2016)

2.2.3.1 Citra Biner

Citra biner adalah citra digital yang hanya memiliki dua warna yaitu hitam dan putih. Setiap piksel dari citra biner hanya direpresentasikan dengan nilai 1 dan 0.

2.2.3.2 Citra Grayscale

Merupakan citra yang merepresentasikan aras keabuan atau intensitas warna putih. Terdapat dua kemungkinan yaitu nilai intensitas paling rendah merepresentasikan warna hitam dan nilai intensitas tinggi merepresentasikan warna putih.

2.2.3.3 Citra Warna

Citra warna mewakili warna yang merupakan kombinasi dasar dari tiga warna yaitu RGB (*Red, Green, Blue*). Dalam citra warna kedalaman warna yang digunakan bergantung pada kedalaman piksel citra tersebut. Citra warna terdiri dari beberapa warna-warna penyusunnya. Setiap warna dasar menggunakan penyimpanan 8bit atau sama dengan 1 *byte*.

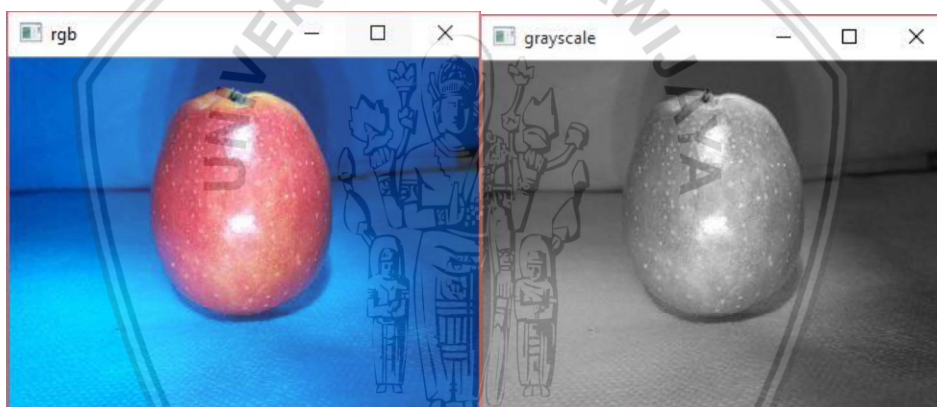
Beberapa *color-space* yang umum menjadi tujuan konversi antara lain (Fitriyah, 2016) :

2.2.2.3.1 Konversi warna dari RGB ke *Grayscale*

Grayscale memiliki 1 nilai intensitas dalam setiap pikselnya. *Grayscale* memiliki warna dari hitam ke putih. Warna hitam memiliki nilai 0 dan putih memiliki nilai 1. **Persamaan 2.1** adalah persamaan matematika untuk mengkonversi nilai RGB ke *Grayscale* adalah:

$$\text{Gray Intensity} = \frac{R+G+B}{3} \quad (2.1)$$

Adapun contoh hasil gambar dari konversi RGB ke *Grayscale* adalah :



Gambar 2.5 Gambar hasil konversi RGB ke Grayscale

2.2.2.3.2 Konversi warna dari RGB ke HSV

HSV (*Hue, Saturaturaion, Value*) umumnya digunakan untuk mendapatkan warna sebenarnya dari sebuah objek. *Hue* dapat memiliki nilai yang berdekatan dari warna yang sama. Sedangkan *Saturation* dan *Value* menunjukkan tingkat luminasi dari sebuah warna **Persamaan 2.2** hingga **Persamaan 2.5** adalah persamaan matematika untuk menghitung nilai RGB ke HSV adalah sebagai berikut :

$$R' = R/255 \quad (2.2)$$

$$G' = G/255$$

$$B' = B/255$$

$$C_{\max} = \max(R', G', B')$$

$$C_{\min} = \min(R', G', B')$$

$$d\Delta = C_{max} - C_{min}$$

Hue calculation

(2.3)

$$H = \begin{cases} 60 \times \left(\frac{G' - B'}{\Delta} \right), & C_{max} = R' \\ 60 \times \left(\frac{B' - R'}{\Delta} \right), & C_{max} = G' \\ 60 \times \left(\frac{R' - G'}{\Delta} \right), & C_{max} = B \end{cases}$$

Saturation calculation

(2.4)

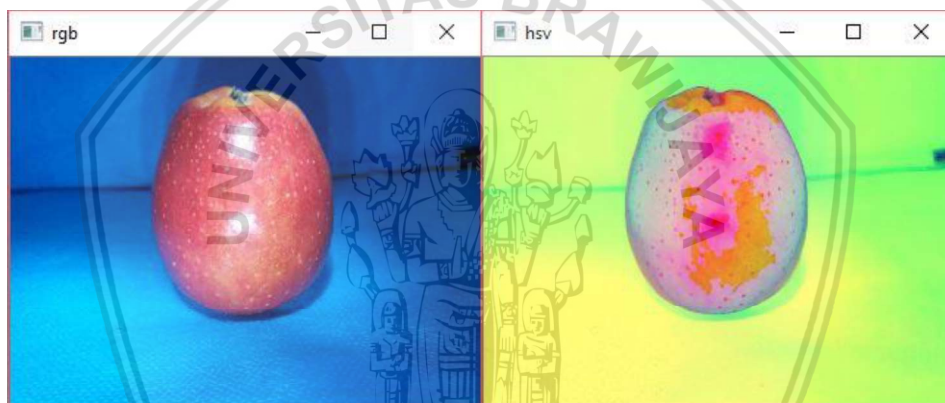
$$S = \begin{cases} 0, & C_{max} = 0 \\ \frac{\Delta}{C_{max}}, & C_{max} \neq 0 \end{cases}$$

Value calculation

(2.5)

$$V = C_{max}$$

Adapun contoh hasil dari konversi RGB ke HSV adalah sebagai berikut



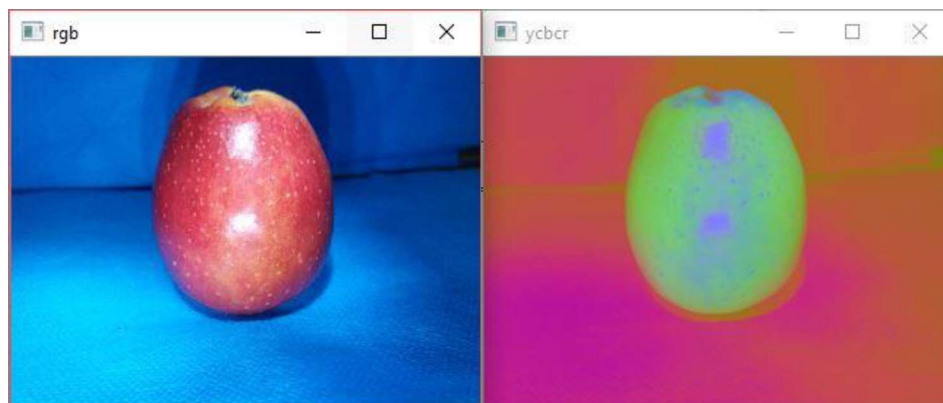
Gambar 2.6 Hasil konversi dari RGB ke HSV

2.2.2.3.3 Konversi warna dari RGB ke YCbCr

Croma (Cb dan Cr) ini digunakan untuk membedakan warna kulit manusia. **Persamaan 2.6** adalah persamaan matematika untuk menghitung nilai RGB ke YCbCr adalah :

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.6)$$

Adapun contoh dari konversi RGB ke YCbCr adalah sebagai berikut :



Gambar 2.7 Hasil konversi dari RGB ke YCbCr

2.2.4 Tekstur

Tekstur mendeskripsikan tentang kehalusan, kekasaran, dan keterturan dalam suatu wilayah tertentu. Dalam pengolahan citra digital tekstur didefinisikan sebagai distribusi spasial dari derajat keabuan di dalam sekumpulan piksel yang bertetangga. Dalam penglihatan manusia, mereka memandang tekstur dalam deksripsi yang berbeda seperti kasar, tidak teratur, teratur, halus dan acak. Hal tersebut merupakan deskripsi non-kumulatif, sehingga untuk memudahkan proses analisis dibutuhkan deskripsi yang kuantitatif (Yudhistira Ganis K, 2015).

2.2.4.1 Gray Level Co-Occurrence Matrix (GLCM)

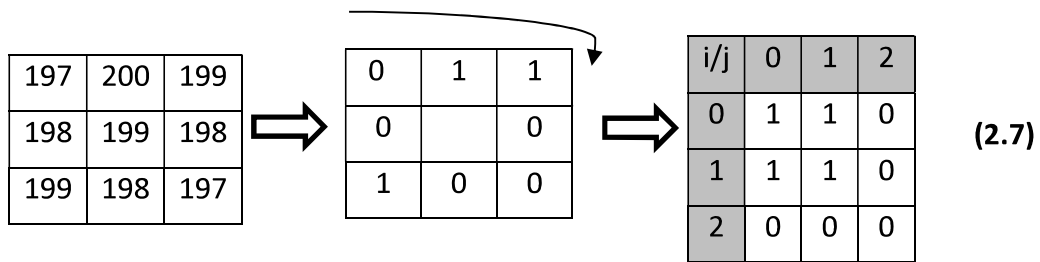
Gray Level Co-Occurrence Matrix adalah salah satu metode yang digunakan untuk menganalisa tekstur. *Co-Occurrence Matrix* atau matrik ko-okurensi dapat dibentuk dari suatu citra dengan melihat piksel yang memiliki intensitas tertentu dan piksel yang berpasangan. Dengan menggunakan metode GLCM untuk mendeteksi tekstur maka ini berdasarkan pada hipotesa bahwa dalam suatu tekstur akan terjadi pasangan aras keabuan atau perulangan konfigurasi. Untuk menganalisa citra berdasarkan distribusi statistik dari intensitas pikselnya, dapat dilakukan dengan mengekstrak fitur teksturnya (Kurniati, 2015).

Pada GLCM terdapat perhitungannya sebagai berikut :

Data matrik berukuran 3x3 dari ukuran sampel GLCM yaitu 30x30 piksel.

$$\text{Nilai matrik} = \begin{bmatrix} 197 & 200 & 199 \\ 198 & 199 & 198 \\ 199 & 198 & 197 \end{bmatrix}$$

Berdasarkan nilai matrik tersebut, dibuat 8 bit angka. Dengan titik pusat berada di nilai tengah dari matrik. Jika nilai lebih besar dan sama dengan dari nilai tengah maka bernilai 1, sedangkan jika nilai lebih kecil dari nilai tengah maka bernilai 0. Dengan $d=1$ dan $\theta = 0^\circ$ sehingga proses *scanning* berjalan dari kiri ke kanan per piksel. **Persamaan 2.7** adalah persamaan matematika untuk mendapatkan nilai biner.



Matrik GLCM diubah menjadi matrik GLCM yang simetris dengan cara ditambah jumlah *transpose* matriknya. **Persamaan 2.8** adalah persamaan matematika untuk mendapatkan nilai matrik GLCM simetris.

Matrik GLCM + *transpose* matrik = Matrik GLCM simetris

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 2 & 0 \\ 2 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (2.8)$$

Berikut adalah nilai matrik yang sudah dinormalisasi yakni membagi tiap elemen dengan jumlah dari elemen matrik.

$$\begin{bmatrix} \frac{2}{8} & \frac{2}{8} & \frac{0}{8} \\ \frac{2}{8} & \frac{2}{8} & \frac{0}{8} \\ \frac{0}{8} & \frac{0}{8} & \frac{0}{8} \end{bmatrix} \quad (2.9)$$

Berdasarkan perhitungan diatas dapat dijelaskan bahwa setelah mendapat nilai matrik yang sudah dinormalisasi dihitung fiturnya untuk diketahui nilainya. Pada penelitian kali ini hanya menggunakan fitur *contrast*.

Tabel 2.1 Fitur GLCM

No	Fitur	Formula
1	Kontras	$\sum_i \sum_j (i - j)^2 C(i, j)$
2	Energi	$\sum_i \sum_j C^2(i, j)$
3	Entropi	$\sum_i \sum_j C(i - j)^2 \log C(i, j)$
4	Homogenitas	$\sum_i \sum_j \frac{C(i, j)}{1 + i + j }$

Nilai matrik = $\begin{bmatrix} \frac{2}{8} & \frac{2}{8} & \frac{0}{8} \\ \frac{2}{8} & \frac{2}{8} & \frac{0}{8} \\ \frac{0}{8} & \frac{0}{8} & \frac{0}{8} \end{bmatrix}$

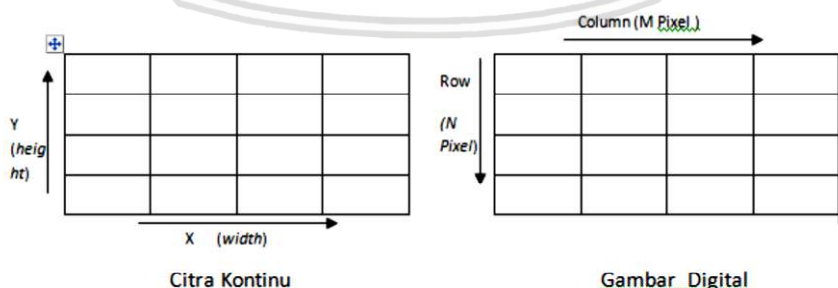
$$Contrast = \sum_i \sum_j (i - j)^2 C(i, j) \quad (2.10)$$

$$\begin{aligned}
 &= (1 - 1)^2 \times \frac{2}{8} + (1 - 2)^2 \times \frac{2}{8} + (1 - 3)^2 \times \frac{0}{8} + \\
 &\quad (2 - 1)^2 \times \frac{2}{8} + (2 - 2)^2 \times \frac{2}{8} + (2 - 3)^2 \times \frac{0}{8} + \\
 &\quad (3 - 1)^2 \times \frac{0}{8} + (3 - 2)^2 \times \frac{0}{8} + (3 - 3)^2 \times \frac{0}{8} \\
 &= 0 + \frac{2}{8} + 0 \\
 &\quad \frac{2}{8} + 0 + 0 \\
 &\quad 0 + 0 + 0 \\
 &= \frac{1}{4} \\
 &= 0.25
 \end{aligned}$$

2.2.5 Ukuran Citra Digital

Ukuran citra digital dapat dinyatakan dalam banyaknya piksel atau titik. Piksel atau *picture element* merupakan bagian terkecil area dari sebuah citra. Dalam sebuah gambar jumlah titik dalam *scanning* disebut juga resolusi. Maka semakin banyak titik dalam suatu citra maka semakin besar pula resolusi dari gambar tersebut. Piksel diatur dalam baris dan kolom. Biasanya baris dapat disimbolkan sebagai N dan kolom dapat disimbolkan sebagai M. Besar dari citra dibatasi oleh jumlah baris dan kolom. Citra digital yang berukuran N dan M dapat dinyatakan pada **Persamaan 2.11** dengan matriks sebagai berikut : (Susanta, 2015)

$$f(x, y) \approx \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M) \\ f(1,0) & f(1,1) & \dots & f(1,M) \\ \vdots & \vdots & \ddots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix} \quad (2.11)$$



Gambar 2.8 Ruang citra kontinyu dan gambar digital
(Susanta, 2015)



Gambar 2.9 Contoh citra dalam piksel yang berbeda
(Susanta, 2015)

2.2.6 Bounding Box

Bounding box merupakan suatu bentuk persegi empat yang melingkupi suatu objek, *bounding box* berfungsi untuk menandai suatu objek yang sudah terdeteksi (Nafidha, 2016). *Bounding box* ini menandai objek sesuai besarnya dengan objek tersebut. (Anton Yudhana, 2016)



Gambar 2.10 Gambar yang sudah di *bounding box*
(Pamungkas, 2016)

2.2.7 Aspectratio

Aspectratio menjelaskan suatu hubungan proporsional antara lebar dan tinggi. Dalam citra ataupun video digital lebar suatu video diukur dari piksel horizontal dari kiri ke kanan. Sedangkan tinggi dari suatu video atau citra diukur dari piksel vertikal dari atas ke bawah. *Aspectratio* tidak memiliki satuan ukuran seperti cm, mm, px, namun secara umum dapat dinyatakan dengan dua angka seperti 16:9, 4:3, 1.771 atau 1.77 (Venantius, 2013) (Nizam, 2015). **Persamaan 2.12** adalah persamaan matematika untuk menghitung *aspectratio*:

$$\text{Aspectratio} = \frac{\text{width}}{\text{height}} \times 100\% \quad (2.12)$$

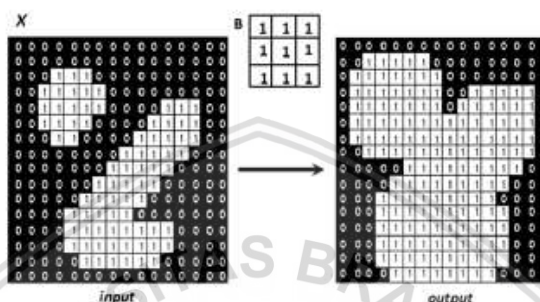
2.2.8 Morphology

Dalam *morphology* terdapat 4 operasi dasar, diantaranya (Fitriyah, 2016):

2.2.8.1 Dilasi

Operasi dilasi yaitu dengan membuat bentuk gambar menjadi terlihat membesar. **Persamaan 2.13** adalah persamaan matematika untuk menghitung nilai dilasi:

$$A \oplus B = \bigcup_{b \in B} A_b \quad (2.13)$$



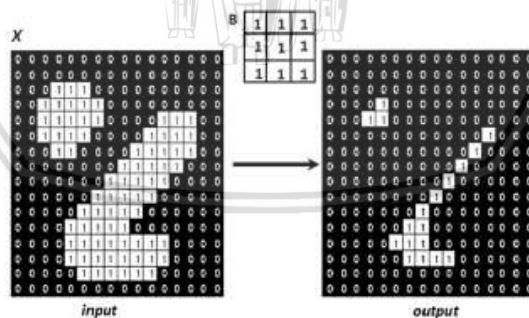
Gambar 2.11 Hasil operasi dilasi

Sumber : (Fitriyah, 2016)

2.2.8.2 Erosi

Erosi merupakan kebalikan dari dilasi, yaitu dengan membuat gambar menjadi terlihat mengecil atau menipis. **Persamaan 2.14** adalah persamaan matematika untuk menghitung nilai erosi:

$$A \ominus B = \bigcap_{b \in B} A_{-b} \quad (2.14)$$



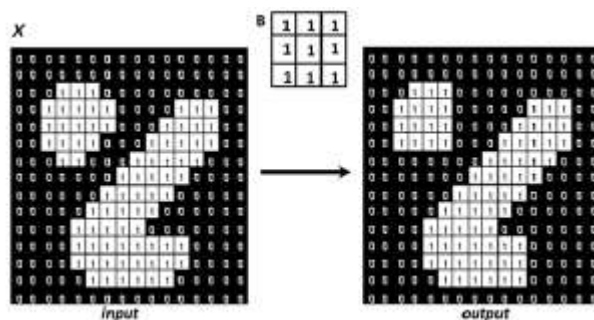
Gambar 2.12 Hasil operasi erosi

Sumber : (Fitriyah, 2016)

2.2.8.3 Opening

Opening dimulai dengan erosi yaitu untuk memperkecil tepi objek. Setelah itu dilakukan proses dilasi yaitu memperbesar segmen citra dengan menambah lapisan di sekitar objek. **Persamaan 2.15** adalah persamaan matematika untuk menghitung nilai *opening*.

$$A \circ B = (A \ominus B) \oplus B \quad (2.15)$$



Gambar 2.13 Hasil operasi *opening*

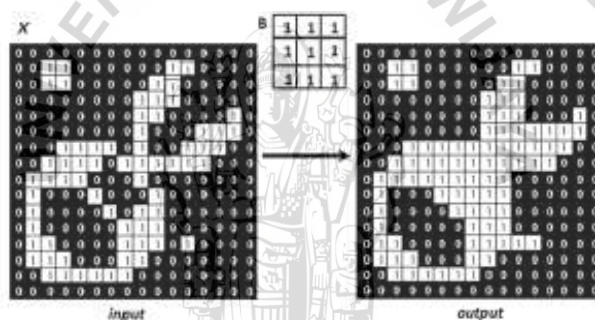
Sumber : (Fitriyah, 2016)

2.2.8.4 Closing

Closing dimulai dengan dilasi yaitu untuk memperbesar segmen citra dengan menambah lapisan di sekitar objek. Setelah itu dilakukan proses erosi yaitu untuk memperkecil tepi objek. **Persamaan 2.16** adalah persamaan matematika untuk menghitung nilai *closing*.

$$A \circ B = (A \oplus B) \ominus B$$

(2.16)

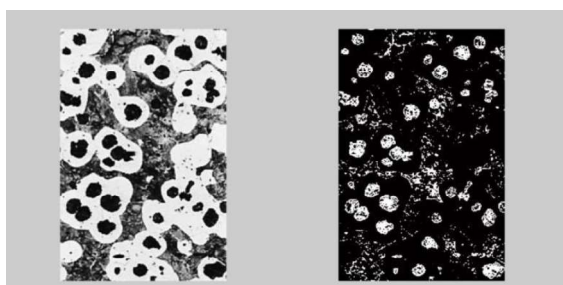


Gambar 2.14 Hasil operasi *closing*

Sumber : (Fitriyah, 2016)

2.2.9 Thresholding

Thresholding merupakan suatu metode segmentasi, dengan memberikan nilai 0 untuk nilai intensitas yang tidak masuk dalam batas sedangkan nilai 1 untuk yang masuk pada batas.



Gambar 2.15 Segmentasi menggunakan *thresholding*

Sumber : (Fitriyah, 2016)

2.2.10 Webcam Logitech

Webcam digunakan untuk menangkap gambar dan merekam video pada objek yang berjalan. Webcam tersebut memiliki resolusi yang besar yaitu 720p untuk merekam video sedangkan untuk memotret memiliki resolusi sebesar 5MP. Webcam memiliki RAM sebesar 2GB, memiliki port USB 2.0 dan memiliki ruang hard drive sebesar 200MB. Untuk menghubungkan antara kamera dengan Raspberry Pi dengan menggunakan kabel USB (Logitech, 2018).



Gambar 2.16 Webcam Logitech

Sumber : (Logitech, 2018)

2.2.11 Raspberry Pi 3

Raspberry Pi merupakan suatu perangkat mini *computer* yang berukuran kecil. Raspberry Pi 3 memiliki *quadcore* CPU 64-bit dan sudah memiliki modul *WiFi* dan *Bluetooth* sehingga lebih memudahkan dalam melakukan komunikasi. Adapun spesifikasi dari Raspberry Pi 3 (MATT, 2016) :

Tabel 2.2 Spesifikasi Raspberry Pi

Spesifikasi	Keterangan
Soc	<i>Broadcom</i> BC 2837 (64-bit ARM Cortex A53)
RAM	1GB
WiFi	802.11n
Bluetooth	<i>Bluetooth Low Energy</i>
GPIO	40pin
Ports	4x USB 2port, CSI camera port, DSI display port, MicroSD card slot, HDMI



Gambar 2.17 Raspberry Pi 3

Sumber : (MATT, 2016)

2.2.12 Motor Servo

Motor Servo adalah sebuah motor DC. Motorservo sendiri terdapat dua jenis yaitu, tipe *standard* dan tipe *continuous*. Jika tipe *standard* hanya dapat bergerak sebesar 180° , sedangkan yang tipe *continuous* dapat bergerak sebesar 360° . Komponen penyusun dari motor servo ini sendiri adalah motor DC, *gearbox*, variabel resistor atau potensio meter dan rangkaian kontrol. Motor servo ini dapat berputar dua arah, mengikuti arah jarum jam atau CW dan berlawanan dengan arah jarum jam atau CCW untuk menggerakkan servo, sudut dari sumbu motor servo diatur berdasarkan lebar pulsa sinyal PWM.

Motor Servo MG995 dapat berputar sebesar 180° dengan tegangan sebesar 4.8V – 7.2V. Untuk menggerakkan motor servo diberi pulsa sebesar 1.5ms nantinya akan bergerak sebesar 90° , jika diberi pulsa lebih dari 1.5ms nantinya akan bergerak sebesar 180° . Pada penelitian ini menggunakan tiga buah sebagai pengatur gerakan lengan pendorong untuk proses sortir buah apel. (Habibi, 2016)

Adapaun spesifikasi dari motor servo MG995 adalah sebagai berikut :

Tabel 2.3 Spesifikasi Motor Servo MG995

Spesifikasi	Nilai
<i>Weight</i>	55g
<i>Dimension</i>	40.7×19.7×42.9mm
<i>Torque</i>	4.8v: 130.54 oz 9.4kg/cm – 6.0v 152.76 oz 11kg.
<i>Operating voltage</i>	4.8v – 6.6v
<i>Gear type</i>	Metal gear
<i>Speed</i>	4.8v: 0.20s/60° 6.6v: 0.16s/60°
<i>Pulse cycle</i>	1ms
<i>Modulation</i>	Digital



Gambar 2.18 Motor Servo MG995

Sumber: (Puspawardhana, 2014)

2.2.13 Motor Stepper

Motor DC memiliki berbagai jenis motor, salah satunya ialah motor stepper. Motor *stepper* ini dikendalikan dengan pulsa-pulsa digital. Motor *stepper* ini bekerja dengan merubah pulsa elektronis menjadi diskrit sehingga pergerakan motor *stepper* ini berdasarkan pulsa yang diberikan dengan terlebih dahulu yang diprogram pada mikrokontroler. Motor *stepper* pada penelitian ini digunakan sebagai penggerak untuk memutar as sehingga sabuk (*belt*) yang prinsip kerjanya seperti konveyor dapat berputar sehingga proses sortir dapat dilakukan. (Habibi, 2016)

Pada penelitian ini menggunakan Motor Stepper Nema 17, dengan spesifikasi sebagai berikut :

Tabel 2.4 Spesifikasi Motor Stepper Nema 17

Spesifikasi	Nilai
Size	42.3 mm square × 48 mm
Weight	350g
Shaft	Single Output
Shaft Diameter	5 mm
Step Angle	1,8°
Holding Torque	3.2 kg cm
Rated Current	1,2 A
Voltage	4 V
Bearing	Ball
Inductance	4,5 ±20%
Lead length	30cm
Steps per revolution	200



Gambar 2.19 Stepper Motor

Sumber : (Inventables, 2017)

2.2.14 Driver Motor A4988

Motor driver ini dapat digunakan untuk mengendalikan stepper motor. Driver ini sudah dilengkapi dengan RESET, STEP, DIR, GND, VDD, VMOT. VMOT digunakan untuk disambungkan dengan motor power supply, sedangkan VDD digunakan untuk logic *power supply*. Pada driver motor ini terdapat pin masukan STEP dan DIR. STEP berfungsi untuk mengatur kecepatan putaran motor stepper dengan diberi sinyal *HIGH* dan *LOW*, sedangkan DIR digunakan untuk mengubah putaran searah jarum jam (CW) atau berlawanan dengan arah jarum jam (CCW). (Nedelkovski, 2018)

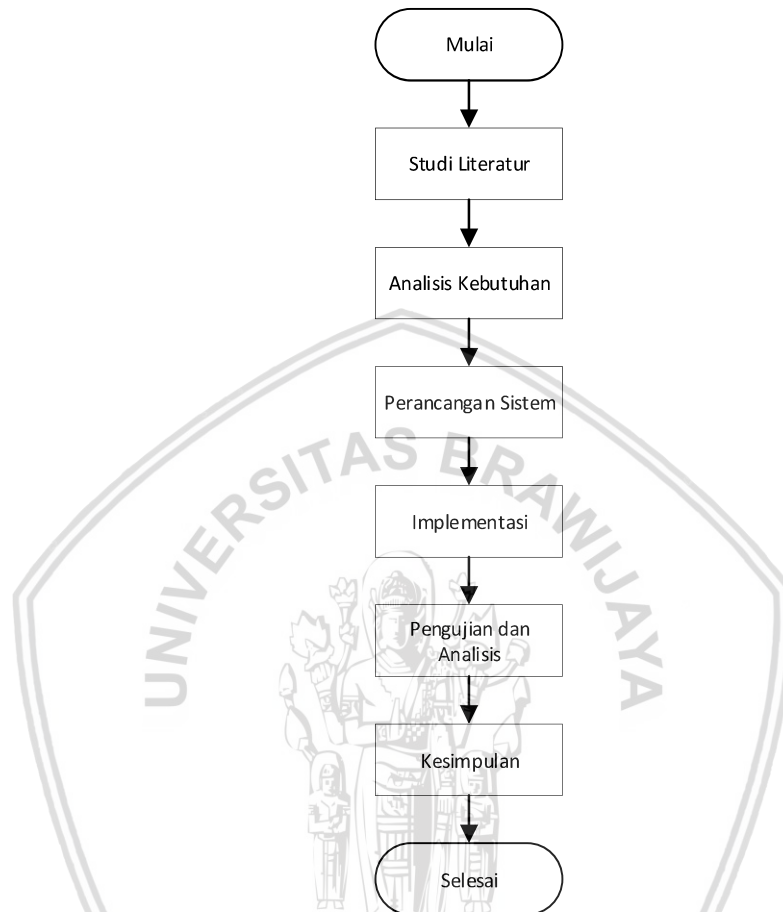


Gambar 2.20 IC A4988

Sumber: (Paulhurleyuk, 2018)

BAB 3 METODOLOGI

Alur metodologi penelitian yang dilakukan secara umum dapat dilihat dari diagram alir berikut ini :



Gambar 3.1 Diagram Alir Metodologi Penelitian

Penelitian dilakukan secara berurut sesuai dengan Gambar 3.1 dimulai dari studi literatur sampai dengan kesimpulan. Pertama yang dilakukan adalah mencari studi literatur atau tinjauan pustaka yang sudah ada dan sesuai dengan topik yang sedang dikerjakan. Analisa kebutuhan dilakukan untuk kelengkapan yang dibutuhkan sebuah sistem. Perancangan sistem adalah kerangka dari sebuah sistem yang akan dibuat. Implementasi adalah tahap percobaan dari sebuah sistem yang sudah dibuat. Pengujian dan analisis dilakukan untuk menguji sistem dan melakukan analisis terhadap performa sistem yang sudah dibuat. Jika semua sistem sudah berjalan sesuai prosedur maka dapat dilakukan penarikan kesimpulan.

3.1 Studi Literatur

Studi Literatur digunakan sebagai teori penguat dan landasan dasar dalam penelitian. Teori pendukung tersebut dari buku, jurnal, skripsi dan internet. Teori yang digunakan adalah teori mengenai :

1. Citra Digital
2. Tekstur
3. GLCM (*Gray Level Co-Occurrence Matrix*)
4. Motor Servo
5. Motor Stepper
6. Opencv
7. Raspberry Pi 3
8. Webcam

3.2 Analisis Kebutuhan

Analisis kebutuhan sistem bertujuan untuk menganalisis semua kebutuhan yang diperlukan oleh sistem yang akan dibangun. Analisis kebutuhan sistem dilakukan dengan mengidentifikasi kebutuhan sistem. Analisis kebutuhan tersebut, meliputi kebutuhan perangkat lunak dan kebutuhan perangkat keras.

3.2.1 Kebutuhan Perangkat Keras

Kebutuhan perangkat keras yang dibutuhkan dalam membangun sistem adalah sebagai berikut:

- a. Laptop
- b. Webcam
- c. Motor Servo
- d. Motor Stepper
- e. Raspberry Pi 3
- f. Mikrokontroler Arduino Uno

3.2.2 Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak yang dibutuhkan dalam membangun sistem adalah sebagai berikut:

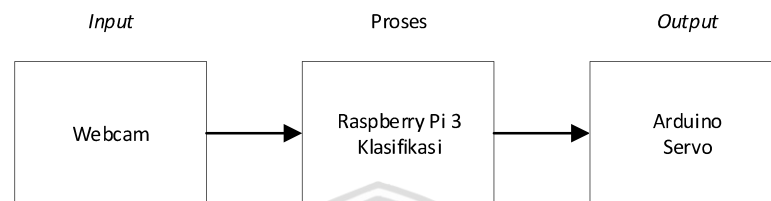
- a. *Operator System* (OS) Windows 10
- b. *Library*
- c. Bahasa Pemrograman Opencv pada Python 3.5

3.3 Perancangan Sistem

Setelah semua kebutuhan didapatkan melalui tahap analisis kebutuhan dilakukan perancangan sistem. Perancangan sistem dibagi menjadi perancangan perangkat keras dan perancangan perangkat lunak.

3.3.1 Perancangan Perangkat Keras

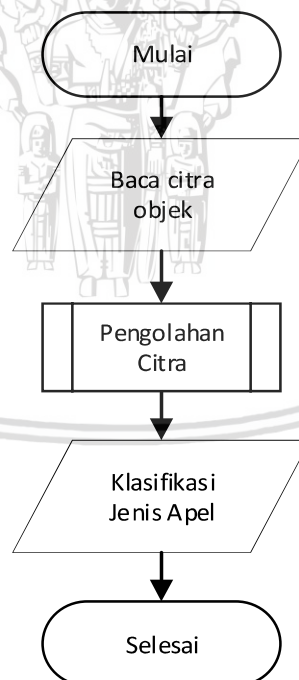
Diagram blok perancangan perangkat keras dapat dilihat pada Gambar 3.2. Dimulai dengan meletakkan apel pada belt konveyor. *Input* berupa buah apel yang dideteksi oleh *webcam* dan dibantu pencahayaan oleh LED yang diletakkan pada atas dan bawah *webcam*. Setelah itu kan mendapatkan hasil yang diproses pada Raspberry Pi , lalu *output* berupa servo akan mendorong buah apel masuk pada jenis klasifikasinya.



Gambar 3.2 Diagram Blok Perancangan Perangkat Keras

3.3.2 Perancangan Perangkat Lunak

Pada tahap ini webcam akan mendeteksi citra objek yaitu berupa kulit buah apel, setelah itu dilakukan proses pengolahan citra, hasil dari pengolahan citra akan mengklasifikasikan buah apel sesuai dengan jenisnya yang ditampilkan pada *compiler* program.



Gambar 3.3 Diagram Blok Perancangan Perangkat Lunak

3.4 Implementasi

Pada tahap implementasi dilakukan berdasarkan perancangan sistem yang telah dibuat. Pada implementasi memberikan desain atau *prototype* sistem yang sudah dibuat dan potongan beberapa program yang telah dibuat.

3.5 Pengujian dan Analisis

Pengujian dan analisis yang diterapkan disini adalah dengan menguji data hasil pengklasifikasian jenis buah Apel dengan beberapa skenario yang dilakukan sebagai berikut :

- a. Pengujian akurasi nilai *aspectratio*.
- b. Pengujian waktu komputasi.
- c. Pengujian ketepatan *hardware* dan *software*.

3.6 Kesimpulan

Pengambilan kesimpulan dilakukan setelah melakukan semua proses pengujian selesai dan mendapatkan hasil yang akurat. Kesimpulan diambil dari hasil pengujian dan analisis terhadap sistem yang dibuat. Pada tahap ini selain diambil kesimpulan, juga diharapkan ada saran untuk melakukan pengembangan lebih lanjut dan memperbaiki sistem. Dapat dilakukan dengan mengganti *hardware*, *software* ataupun metode yang digunakan agar lebih baik lagi.



BAB 4 REKAYASA KEBUTUHAN

4.1 Gambaran Umum Sistem

Klasifikasi Jenis Buah Apel Lokal Berdasarkan Penciri Warna, *Aspectratio* dan GLCM Menggunakan *Belt* Konveyor Berbasis Raspberry Pi merupakan suatu sistem klasifikasi dengan parameter berupa bentuk, warna dan tekstur dari kulit buah apel lokal. Sistem dibuat dan memanfaatkan konveyor berjalan seperti pada pabrik-pabrik dengan meletakkan kamera *webcam* di samping pintu masuk apel dan juga menambahkan *LED* agar proses pendeteksian bentuk, warna, dan tekstur lebih maksimal. Parameter yang digunakan dalam menentukan jenis apel nantinya adalah nilai *Hue* dalam HSV, *aspectratio* dan nilai yang didapat dari GLCM *Contrast*. Raspberry Pi 3 digunakan sebagai mini komputer untuk memproses pengolahan citra. Dalam penelitian ini juga menggunakan Arduino yang berfungsi sebagai penggerak Motor *Stepper*, Motor *Servo* dan sabuk (*belt*) *conveyor*. Raspberry Pi 3 dan Arduino berkomunikasi dengan menggunakan Serial I2C sehingga keduanya dapat saling mengirim dan menerima data. Pada conveyor terdapat pendorong yang berfungsi melakukan klasifikasi jenis apel dengan menggunakan *rule evaluation*.

4.1.1 Tujuan

Tujuan dari dibuatnya sistem ini adalah untuk memudahkan melakukan sortir buah apel berdasarkan jenis buahnya guna menghindari perbedaan persepsi saat dilakukan sortir manual. Adapun jenis apel yang digunakan ada 4 jenis yaitu apel ana, manalagi, romebeauty dan wanglin.

4.1.2 Karakteristik Pengguna

Karakteristik pengguna yang dapat menggunakan sistem ini adalah semua orang, tidak ada batasan umum untuk menggunakannya, tetapi jika anak kecil kurang dari 12 tahun hendaknya didampingi orang dewasa. Laki-laki atau perempuan diperbolehkan menggunakan sistem ini, fisik tidak berpengaruh dalam menggunakan sistem ini karena pengguna cukup menekan tombol *Run*.

4.1.3 Lingkungan Operasi

Persyaratan kebutuhan lingkungan operasi yang mendukung sistem ini yaitu :

1. *Hardware* diletakkan di dalam ruangan, sehingga semua proses dilakukan di dalam ruangan dengan cahaya yang terang.
2. *Webcam* diletakkan menghadap ke depan pada salah satu sisi konveyor, atas dan bawah dari *webcam* ditambahkan *LED* untuk memaksimalkan proses klasifikasi.
3. *Webcam* harus terhubung pada Raspberry Pi 3 untuk dilakukan proses pengolahan citra.

4. Guna mengerakkan *belt* maka Motor Stepper harus tersambung dengan *driver* dan Arduino Uno dengan kecepatan yang stabil yang sudah diatur pada program pada IDE Arduino.

4.1.4 Batasan Perancangan dan Implementasi

Adapun Batasan perancangan dan implementasi dari sistem ini adalah sebagai berikut :

1. Hanya menggunakan 1 buah webcam yang diletakkan pada salah satu sisi konveyor.
2. Menggunakan 4 jenis buah apel yaitu apel ana, manalagi, romebeauty dan wanglin.
3. Terdapat 3 lengan pendorong sehingga jenis apel ana akan di dorong oleh aktuator 1, apel manalagi oleh aktuator 2, apel romebeauty oleh aktuator 3 sedangkan apel wanglin berjalan lurus mengikuti arah konveyor.
4. Hasil klasifikasi hanya dapat dilihat pada *compiler* program, sedangkan aktuator akan bergerak bergantian sesuai nilai yang didapat.
5. Guna menghubungkan semua perangkat dengan menggunakan kabel.
6. Raspberry Pi 3 dan Arduino Uno saling berkomunikasi untuk menjalankan servo/aktuator dengan menggunakan komunikasi I2C.
7. Diperlukan catu daya yang cukup untuk menjalankan konveyor.
8. *Belt* konveyor berwarna biru polos tanpa tekstur.

4.1.5 Asumsi dan Ketergantungan

Asumsi dan ketergantungan dari sistem ini adalah :

1. Seluruh pin dihubungkan pada tiap-tiap komponen sesuai dengan fungsi pada *datasheet*.
2. Posisi webcam tidak boleh berubah dan diberi LED di atas dan bawah webcam untuk memaksimalkan proses pendeteksian citra.

4.2 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem dilakukan untuk menggali semua kebutuhan yang diperlukan untuk Sistem Klasifikasi Jenis Apel ini. Dalam melakukan analisis kebutuhan sistem terdiri atas beberapa kebutuhan yang perlu dijabarkan yakni kebutuhan fungsional dan kebutuhan nonfungsional, dimana kebutuhan non fungsional terdiri dari kebutuhan perangkat keras dan kebutuhan perangkat lunak.

4.2.1 Kebutuhan Fungsional

Berikut ini adalah kebutuhan fungsional yang harus mampu dilakukan oleh sistem:

1. Kamera Webcam

Kamera *Webcam* digunakan untuk mengambil citra kulit buah. Webcam diletakkan pada salah satu sisi dan bersifat tetap. Webcam akan mendeteksi salah satu sisi permukaan kulit buah apel untuk selanjutnya di proses pada *opencv* di Raspberry Pi 3.

2. Parameter pendeteksian

Hasil dari webcam berupa citra buah apel lalu diolah pada perangkat lunak *opencv* dengan 3 buah parameter yaitu berupa nilai *Hue* dari format warna HSV dan *aspectratio* dimana terdapat nilai tinggi (*Heigh*) dan lebar (*Width*). Sedangkan untuk mendapatkan hasil tekstur kulit Apel menggunakan metode *Grey Level Co-Occurrence Matrix* (GLCM).

3. Raspberry Pi 3 dapat mengirim data ke Arduino untuk menggerakkan Motor Servo

Raspberry Pi 3 akan mengirim data ke Arduino yang sudah saling tersambung dengan Pin GPIO pada kedua *device* dengan menggunakan I2C. Setelah kedua *device* tersambung maka Raspberry Pi 3 akan mengirim data ke Arduino untuk selanjutnya diproses pada Arduino. Pada Arduino ini Motor Servo akan bergerak sesuai perintah dari Raspberry Pi 3.

4. Data dari pengolahan citra selanjutnya diproses untuk melakukan proses pengklasifikasian

Data nilai yang diperoleh adalah nilai *Hue* dalam HSV, *aspectratio* dan nilai dari *Grey Level Co-Occurrence Matrix* (GLCM) selanjutnya diproses menggunakan *rule evaluation* untuk dilakukan proses klasifikasi sesuai dengan jenis apel.

4.2.2 Kebutuhan Non Fungsional

Kebutuhan non fungsional dari sistem ini terdiri dari kebutuhan perangkat keras dan kebutuhan perangkat lunak yang dijelaskan secara rinci dibawah ini.

4.2.2.1 Kebutuhan Perangkat Keras

Guna mendukung implementasi pembuatan sistem dari sisi perangkat keras maka di perlukan beberapa alat yang dijelaskan berikut ini:

1. Raspberry Pi 3

Raspberry Pi 3 digunakan sebagai otak untuk melakukan pengolahan informasi. Untuk proses komunikasi data menggunakan WiFi yang sudah terdapat pada Raspberry Pi 3.

2. Kamera Webcam

Webcam digunakan untuk menangkap objek yang berjalan dengan format video.

3. Motor Stepper

Stepper berfungsi sebagai penggerak untuk memutar as sehingga sabuk (*belt*) pada konveyor dapat berputar.

4. Motor Servo

Servo digunakan sebagai penggerak pintu yang digunakan untuk proses klasifikasi jenis buah apel. Pintu disini dimaksud sebagai pendorong yang akan mendorong buah apel masuk ke wadah.

5. Kabel Jumper

Kabel jumper digunakan untuk menghubungkan antar komponen pada alat yang sudah dibuat.

6. Mikrokontroler Arduino Uno

Arduino uno digunakan sebagai penggerak dari motor servo untuk mendorong buah apel masuk pada klasifikasi yang sesuai. Mikrokontroler Arduino juga digunakan untuk menggerakkan motor stepper untuk memutar as sehingga sabuk (*belt*) pada konveyor dapat berputar.

7. LED

Light Emitting Diode (LED) digunakan sebagai pencahayaan yang diletakkan di atas dan bawah webcam.

8. Laptop

Laptop disini berfungsi sebagai media untuk membuat sebuah program agar proses pendeteksian objek, proses berjalannya motor dapat dikendalikan sesuai kebutuhan. Selain itu juga berfungsi sebagai pemberi sumber daya pada mikrokontroler agar dapat berfungsi dengan baik. Adapun spesifikasi Laptop yang digunakan adalah sebagai berikut :

- Model Perangkat : DELL Inspiron 14
- Prosesor : Intel Core i3-5005U CPU 2,0 GHz
- Sistem Operasi : Windows 10 64-bit

4.2.2.2 Kebutuhan Perangkat Lunak

Pada bagian kebutuhan perangkat lunak yang dibutuhkan pada penelitian ini adalah Python 3.5 untuk menjalankan program pada proses klasifikasi buah apel dengan menggunakan beberapa library yaitu library "`matplotlib.pyplot as plt`" adalah library plotting 2D Python yang dapat digunakan untuk membuat plot. Library "`numpy as np`" digunakan untuk operasi matematika diantaranya operasi vector dan matriks. Penggunaan "`skimage.feature`" digunakan untuk menghitung GLCM dengan melakukan import pada "`greycomatrix`" sedangkan "`greycoprops`" digunakan untuk menghitung properti pada GLCM. Untuk memanggil library cv2 digunakan "`import cv2`". Untuk menghitung waktu komputasi digunakan "`import time`". Untuk melakukan perhitungan

metematika digunakan library `"import math"`. Dan yang terakhir library SMBus digunakan untuk komunikasi serial dengan menggunakan I2C.

4.3 Pengumpulan Data

Pengumpulan data dilakukan dengan mengambil data sample dari keempat jenis apel. Pengambilan dilakukan sebanyak 5 kali untuk masing-masing jenis, sehingga terdapat 20 sample. Dari sample tersebut diambil data berupa nilai *Hue*, *aspectratio* dan *GLCM Contrast*.



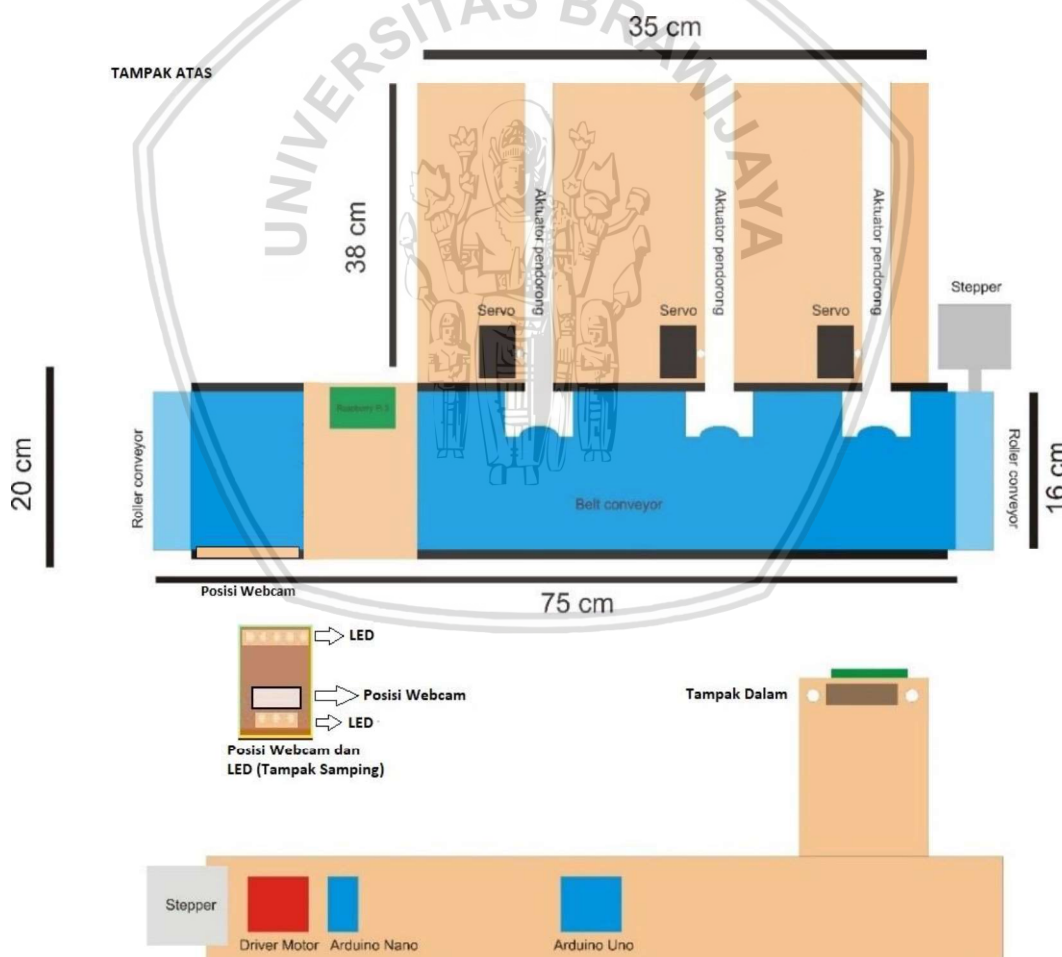
BAB 5 PERANCANGAN DAN IMPLEMENTASI

5.1 Perancangan Sistem

Dalam sub-bab ini akan menjabarkan cara perancangan sistem dimulai dari perancangan *prototype* alat, perancangan perangkat keras hingga perancangan perangkat lunak.

5.1.1 Perancangan *Prototype* Klasifikasi Buah Apel

Dalam melakukan perancangan Sistem Klasifikasi Buah Apel lokal perlu dilakukan desain *prototype* terlebih dahulu. Pembuatan desain *prototype* dilakukan dengan menggunakan aplikasi *CorelDraw* untuk menggambar desain alat yang berbentuk *conveyor* dengan 3 buah lengan yang dapat mendorong buah apel untuk melakukan proses klasifikasi, dan tentunya terdapat satu buah kamera yang posisinya tetap disamping alat. *Prototype* dapat dilihat pada Gambar 5.1.



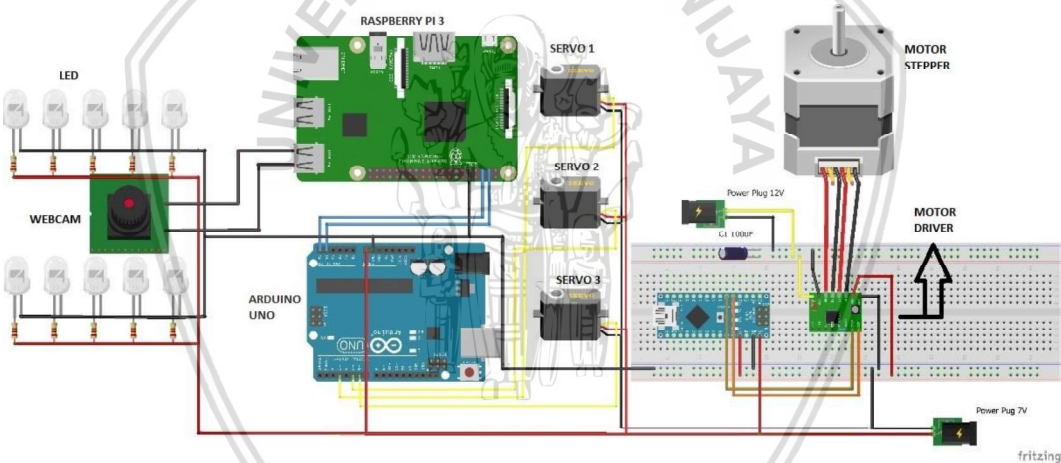
Gambar 5.1 Desain Prototype Alat Klasifikasi Buah Apel

Sistem sortir buah apel berbentuk *conveyor* berjalan. Sabuk *conveyor* digulung pada sebuah as yang diputar dengan Motor *Stepper* dengan *shaft coupling* sebagai penghubung. Pada conveyor juga terdapat sebuah Webcam yang

terdapat pada salah satu sisi conveyor yang bersifat tetap. Pada sekitar Webcam terdapat LED yang digunakan untuk mengurangi bayangan dari buah apel.

5.1.2 Perancangan Perangkat Keras

Perancangan perangkat keras dilakukan berdasarkan analisis kebutuhan perangkat keras serta spesifikasi dari masing-masing dari perangkat keras agar dapat membangun sistem sesuai dengan yang diharapkan. Diagram blok yang sudah dijelaskan pada Gambar 3.2 merupakan gambaran secara umum, sedangkan pada tahap perancangan perangkat keras ini akan dijelaskan secara rinci hubungan perangkat keras masing-masing pin yang digunakan. Webcam yang dihubungkan dengan Raspberry Pi 3 digunakan untuk inputan citra. Disekitar webcam terdapat LED yang dihubungkan pada *Ground* Arduino juga pada *project board*. Raspberry Pi 3 digunakan sebagai pengolahan citra hasil hingga mendapatkan hasil untuk proses penentuan jenis apel. Tiga buah motor servo disini digunakan sebagai penggerak lengan pendorong yang dihubungkan pada Arduino dan *project board*. Untuk menggerakkan *belt conveyor* menggunakan Motor Stepper yang dihubungkan dengan *driver* motor A4988. Skematik rangkaian perancangan perangkat keras dapat dilihat pada Gambar 5.2 :



Gambar 5.2 Skematik Rangkaian Perangkat Keras

Pada skematik diatas masing-masing komponen akan dikaitkan antar pin yang selanjutnya akan dijelaskan pada Tabel 5.1 sampai dengan Tabel 5.7

Tabel 5.1 Pin antara *Motor Driver* dengan *Power Supply*

Pin <i>Motor Driver</i>	Power Supply
V_{Mod}	12 V
GND	GND

Pada Tabel 5.1 menghubungkan antara pin dari *Motor Driver* dengan *Power Supply*. Pada *Motor Driver* terdapat dua pin yang akan disambungkan pada *power supply*. Pin V_{Mod} disambungkan pada 12V dan pin GND disambungkan pada GND *Power Supply* untuk memberi catu daya.

Tabel 5.2 Pin antara Motor *Stepper* dengan Motor *Driver*

Pin Motor <i>Stepper</i>	Pin Motor <i>Driver</i>
2A	2A
2B	2B
1A	1A
1B	1B

Pada Tabel 5.2 menghubungkan antara Motor *Stepper* dan Motor *Driver*. Terdapat 4 pin yang masing-masing akan dihubungkan. Pin 2A pada Motor *Stepper* akan dihubungkan pada pin yang sama pada Motor *Stepper*. Pin 2B akan dihubungkan pada pin yang sama yaitu Pin 2B pada Motor *Driver*. Pin yang ketiga yaitu, pin 1A pada Motor *Stepper* akan dihubungkan pada pin 1A pada Motor *Driver*. Dan yang terakhir pin 1B dihubungkan pada pin yang sama juga pada Motor *Driver*.

Tabel 5.3 Pin antara Motor *Driver* dengan Arduino Uno

Pin Motor <i>Driver</i>	Pin Arduino Uno
V _{DD}	5 V
GND	GND
STEP	D10
DIR	D9

Pada Tabel 5.3 menghubungkan antara V_{DD} Motor *Driver* dengan 5V. Lalu pin GND disambungkan pada GND Arduino Uno. Setelah itu pin STEP dihubungkan pada pin digital Arduino Uno yaitu pin D10 dan Pin DIR dihubungkan pada pin digital D9, penghubungan kedua pin tersebut bertujuan untuk mengatur arah putar dan kecepatan putaran dari Motor *Stepper*.

Tabel 5.4 Pin antara GPIO Raspberry Pi 3 dengan Arduino Uno

Pin GPIO Raspberry	Pin Arduino Uno
GPIO 3 (I2C 1 SDA)	A4 (SDA)
GPIO 5 (I2C 1 SCL)	A5 (SCL)

Tabel 5.4 ini menghubungkan antara Pin GPIO Raspberry Pi 3 dengan Arduino Uno. Terdapat dua pin komunikasi serial SDA dan SCL, SDA (Serial Data) merupakan data serialnya sedangkan SCL (*Serial Clock*) merupakan jalur clock sinkronisasinya. Pin GPIO 3 pada Raspberry Pi 3 dihubungkan pada pin A4 SDA sedangkan pada GPIO 5 dihubungkan pada pin A5 SCL. Fungsi dari kedua pin tersebut dihubungkan adalah saling mengirim dan menerima data. Kedua pin tersebut dihubungkan agar nantinya Raspberry Pi 3 dapat mengirim perintah untuk menjalankan Motor Servo sesuai dengan jenis klasifikasi jenis apel.

Tabel 5.5 Koneksi antara Motor Servo dengan Power Supply

Pin Motor Servo	Pin Power Supply
VCC servo 1	7 V
GND servo 2	GND
VCC servo 2	7 V
GND servo 2	GND
VCC servo 3	7 V
GND servo 3	GND

Tabel 5.6 Pin antara Motor Servo dengan Arduino Uno

Pin Motor Servo	Pin Arduino Uno
Pin signal servo 1	D3
Pin signal servo 2	D5
Pin Signal servo 3	D6

Pada Tabel 5.5 menghubungkan Motor Servo dengan *Power Supply* masing-masing VCC dan GND saling dihubungkan dengan sumber tegangan sebesar 7V dari *Power Supply*. Sedangkan pada Tabel 5.6 menghubungkan tiap-tiap Motor Servo dengan Pin D3, D5 dan D6, lalu VCC dan GND dari ketiga Motor Servo diparalel pada Arduino Uno.

Tabel 5.7 Menyambungkan Webcam dengan Raspberry Pi 3

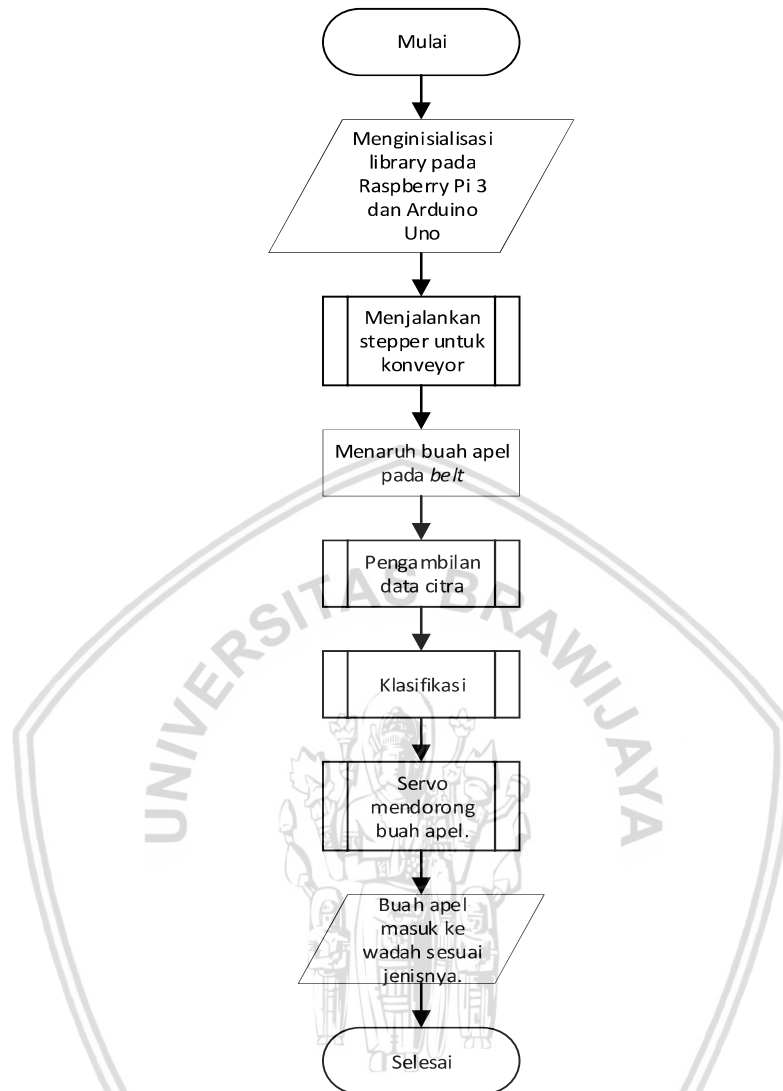
Webcam Logitech	Raspberry Pi 3
<i>Webcam USB Cable</i>	USB

Pada Tabel 5.7 kabel USB pada Webcam dihubungkan pada port USB pada Raspberry Pi 3 untuk menangkap citra kulit buah apel.

5.1.3 Perancangan Perangkat Lunak

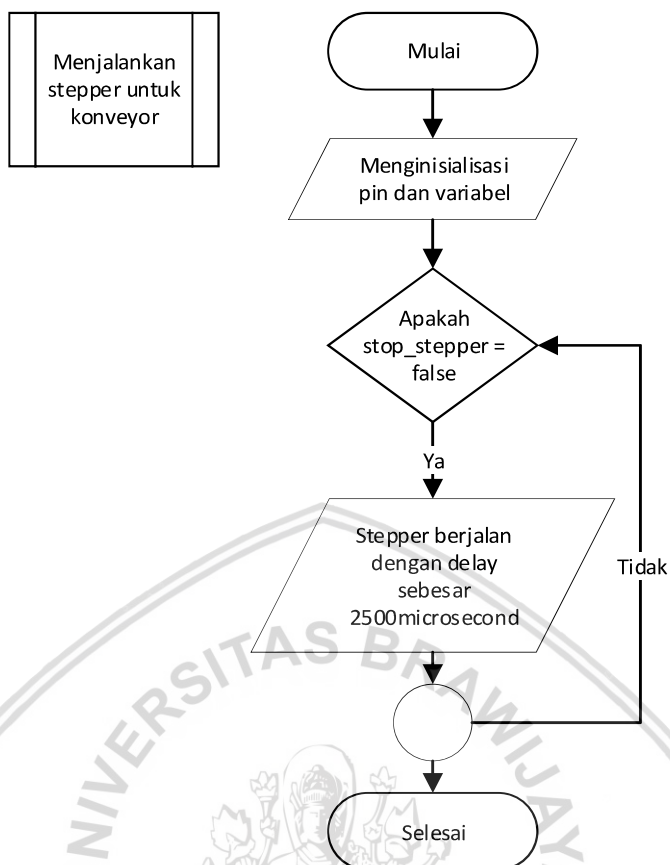
Pada subbab Perancangan Perangkat Lunak ini terbagi menjadi dua pembahasan, yaitu perancangan perangkat lunak pada mikrokontroler untuk melakukan proses pengambilan data yang nantinya akan diolah serta perancangan perangkat lunak untuk melakukan proses klasifikasi.

5.1.3.1 Perancangan Program Utama Sistem



Gambar 5.3 Flowchart Program Utama Sistem

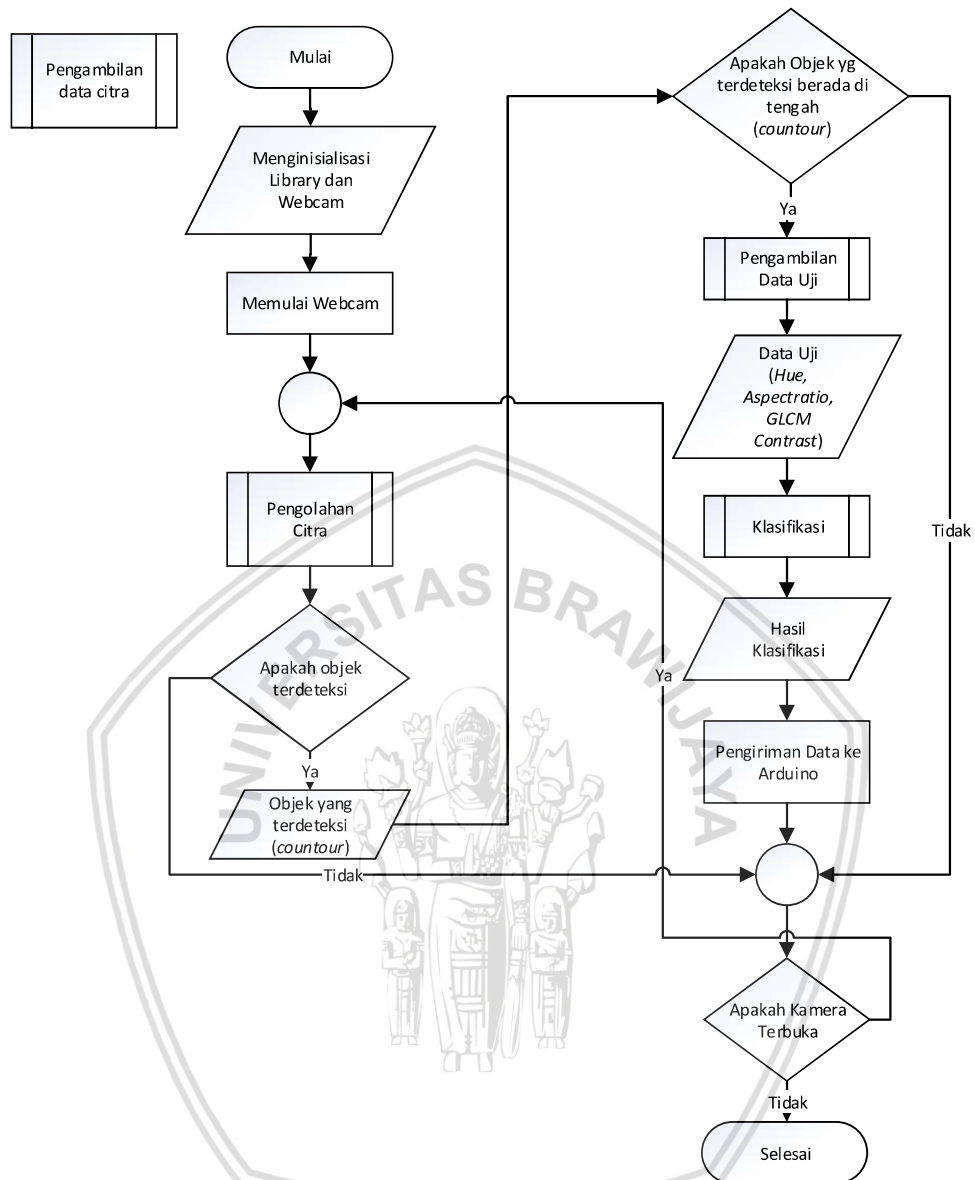
Pada implementasi klasifikasi jenis apel awalnya diperlukan inisialisasi *library* pada *opencv* dan *IDE Arduino*. Diperlukan catu daya untuk menjalankan motor *stepper*. Setelah itu buah apel yang akan di klasifikasi ditaruh pada *belt* dengan jarak yang sudah ditentukan dari *webcam*. Webcam akan mengambil citra kulit buah dengan beberapa tahapan yang akan di lakukan. Hasil dari pengolahan citra adalah berupa nilai *Hue* dari HSV, GLCM *Contrast*, dan *Aspectratio*. Nilai-nilai tersebut akan dijadikan masukan untuk melakukan klasifikasi. Dilakukan komunikasi serial I2C antara Raspberry Pi 3 dengan Arduino Uno pada SDA dan SCL. Fungsi dari kedua pin tersebut adalah untuk saling mengirim dan menerima data. Setelah itu buah akan masuk pada wadah sesuai jenis klasifikasinya.



Gambar 5.4 Flowchart Stepper

Pada gambar diatas merupakan diagram alir untuk menjalankan Motor *Stepper*. Dimulai dengan inisialisasi pin dan variable pada IDE Arduino. Pada saat dihubungkan dengan catu daya *stepper* dalam keadaan berjalan. Setelah itu Raspberry Pi mengirim nilai ke Arduino yang disimpan dalam variable berupa *number*, jika *number* bernilai 5 maka stepper akan berjalan, jika *number* bernilai 4 maka stepper akan berhenti. Penggunaan angka 4 dan 5 hanya berupa variabel yang bisa diganti sesuai kebutuhan. Motor stepper akan berjalan dengan diberi *delay* sebesar 2500 *microsecond*. Program akan terus berjalan sesuai kondisi yang diberikan oleh *user*.

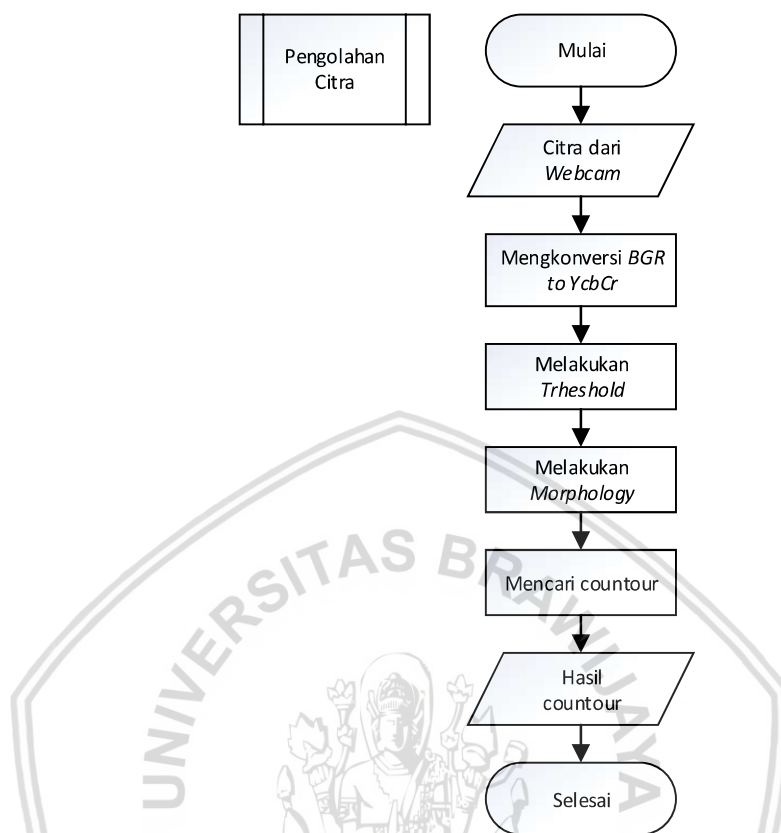
5.1.3.2 Perancangan Pengambilan Data Citra



Gambar 5.5 Flowchart Pengambilan Data Citra

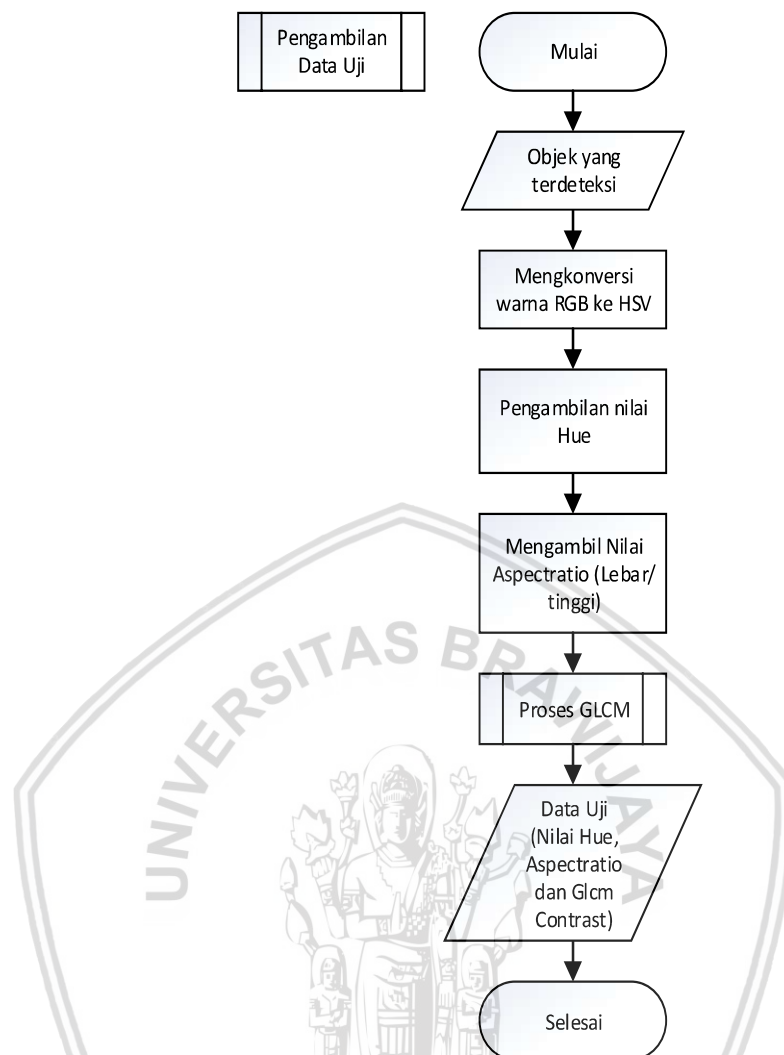
Pada Gambar 5.5 merupakan alur untuk pengambilan data citra. Dimulai dengan inisialisasi *library*, lalu memulai webcam. Pada tahap pengolahan citra, terdapat proses yang dilakukan seperti *Morphology*, *Threshold* dan beberapa tahapan yang nantinya akan dijelaskan secara terpisah. Pada *frame* dibuat garis tengah yang berfungsi untuk mendeteksi objek. Jika objek terkena garis tersebut maka sistem akan mendeteksi ada sebuah objek. Objek sudah terdeteksi dengan webcam dan posisi berada di tengah, terdapat 3 buah kotak kecil berukuran 30x30 yang digunakan untuk mengambil *sample* untuk proses GLCM dan 1 kotak yang digunakan untuk mengambil *sample* warna *Hue*. Setelah itu akan didapat nilai dari *Hue*, GLCM Contrast dan *Aspectratio*. Dari ketiga nilai tersebut akan diklasifikasi dengan menggunakan *rule evaluation* atau percabangan *if*. Setelah itu pada *compiler* Opencv terdapat keluaran berupa tulisan jenis apel apa. Dan saat itu juga

Raspberry Pi 3 mengirim data ke Arduino untuk menggerakkan servo guna menggerakkan lengan untuk memasukkan apel ke jenisnya.



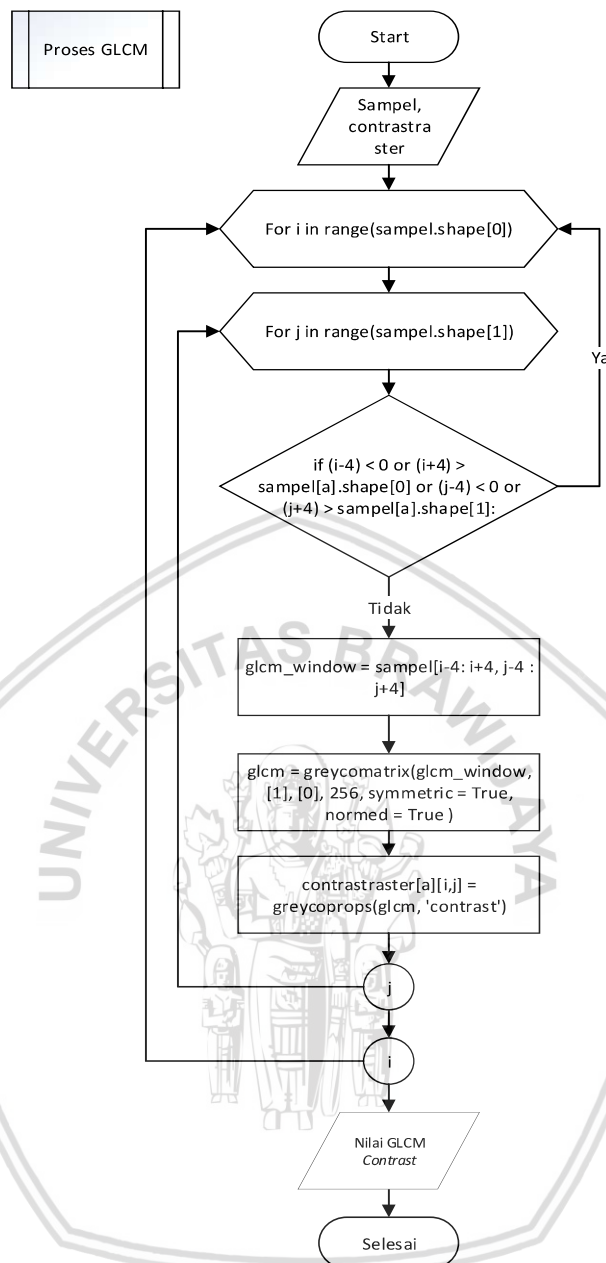
Gambar 5.6 Flowchart Pengolahan Citra Apel

Pada Gambar 5.6 adalah proses untuk pengolahan data citra yang ditangkap oleh webcam berupa video. Setelah itu format warnanya dirubah dari BGR ke *YcbCr* yang digunakan untuk memisahkan *background* dan *foreground*. Setelah itu akan ditentukan *Threshold* untuk menentukan batas atas dan batas bawah. Setelah ditentukan nilai *Threshold* maka dilakukan salah satu proses *Morphology* yaitu *opening*, dilakukannya *opening* ini bertujuan untuk memperbaiki hasil segmentasi citra. Proses *opening* dimulai dengan erosi yaitu untuk memperkecil tepi objek. Setelah itu dilakukan proses dilasi yaitu memperbesar segmen citra dengan menambah lapisan di sekitar objek. Proses *opening* dilakukan untuk menghilangkan objek-objek kecil untuk menjadikan tepian citra menjadi halus agar citra masukan menjadi bentuk yang diinginkan. Setelah itu dilakukan proses kontur untuk menentukan area dalam objek yang akan dideteksi. Hasil dari pengolahan citra berupa nilai dari *Hue*, *Aspectratio* dan nilai *GLCM Contrast*.



Gambar 5.7 Flowchart Pengambilan Data Uji

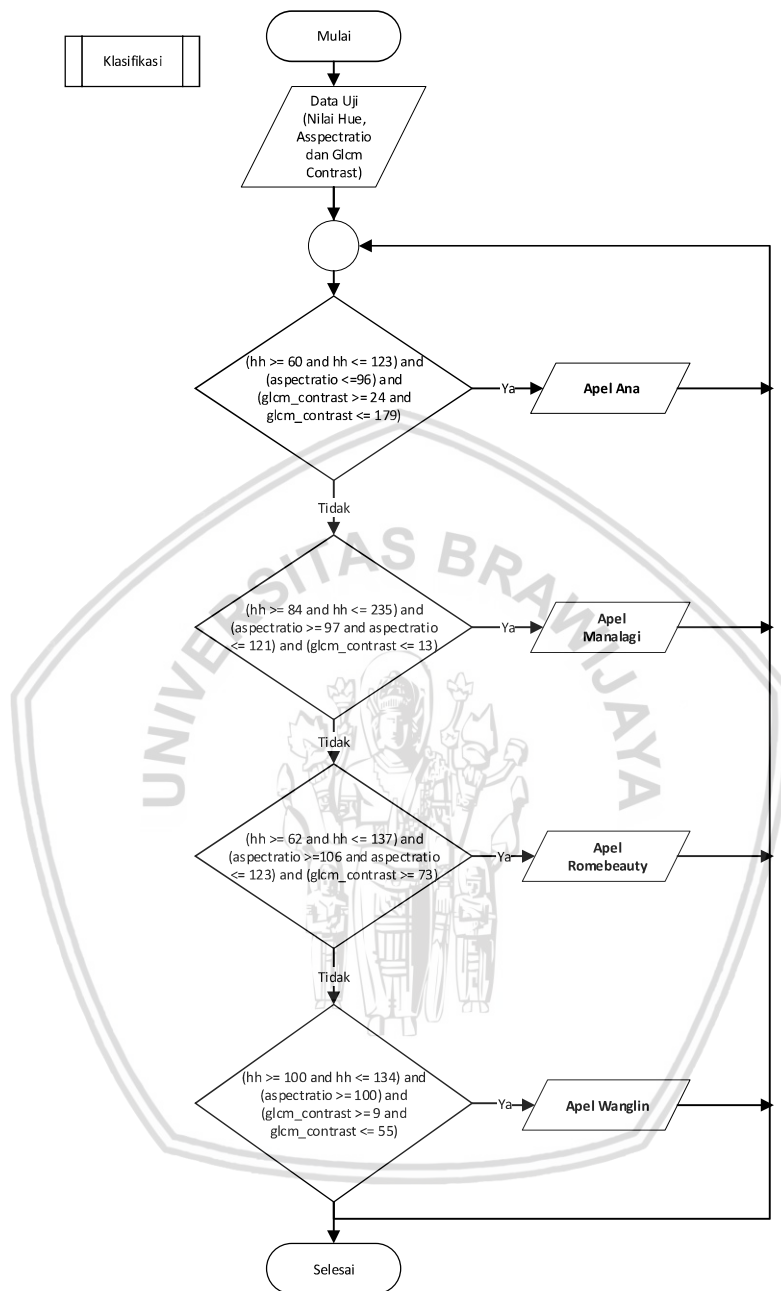
Pada Gambar 5.7 ini, objek yang sudah terdeteksi akan diambil sampel untuk dilakukan konversi warna dari RGB ke HSV. Pada konversi ini hanya mengambil nilai *Hue* saja untuk menentukan *redness* dan *greenness*. Setelah itu untuk menentukan jenis apel bulat atau lonjong dengan menggunakan *aspectratio* dengan perhitungan tinggi dan lebar dari piksel apel. Perhitungan tinggi dimulai dari piksel 0 sampai piksel akhir yang berada di bawah, sedangkan untuk menentukan lebar piksel apel dimulai dari titik 0 sebelah kiri sampai titik akhir citra di bagian kanan. Jika nilai tinggi dan lebar diperoleh lalu dibagi dua untuk mendapatkan hasil jenis apakah apel tersebut. Selanjutnya dilakukan proses GLCM dengan mengambil titik piksel ke 0 lalu dari piksel ke 0 diperiksa empat piksel ke atas, bawah, kiri dan kanan untuk dilakukan identifikasi. GLCM memperoleh ciri dari nilai piksel matrik yang mempunyai nilai tertentu dan membentuk suatu pola tertentu. Fitur yang digunakan dalam GLCM ini ialah kontras. Fitur kontras memiliki penyebaran elemen matrik pada citra yang terletak jauh dari diagonal utama sehingga akan memiliki nilai kontras yang cukup besar. Setelah ketiga tahap diatas dilalui, maka diketahui nilai *Hue*, *Aspectratio* dan GLCM *Contrast*.



Gambar 5.8 Flowchart Proses GLCM

Pada Gambar 5.8 dilakukan proses GLCM, dimulai dengan inputan berupa sampel yang akan dicari nilai GLCM Contrast nya. Lalu dilakukan proses *scanning*, untuk baris disimbolkan dengan i , sedangkan untuk kolom disimbolkan dengan j . Proses ini dilakukan dari baris 1 lalu bergerak ke kanan hingga kolom terakhir. setelah proses selesai, maka ganti baris yaitu baris ke 2 hingga kolom terakhir untuk dilakukan *scanning*. Proses ini terus berlanjut hingga baris terakhir. Untuk kolom maka dilakukan proses yang sama seperti baris yaitu, dilakukan *scanning* pada kolom 1 lalu bergerak ke bawah hingga baris terakhir, selanjutnya kolom 2 hingga baris terakhir. `glcm_windows` adalah sebuah kotak berukuran 8x8 yang digunakan untuk *scanning* pada gambar yang dapat bergerak mengikuti piksel

keberapa untuk *scanning*. Setelah itu dilakukan proses penghitungan nilai glcm texture . Setelah itu ditampilkan hasil pada compiler yaitu nilai GLCM.

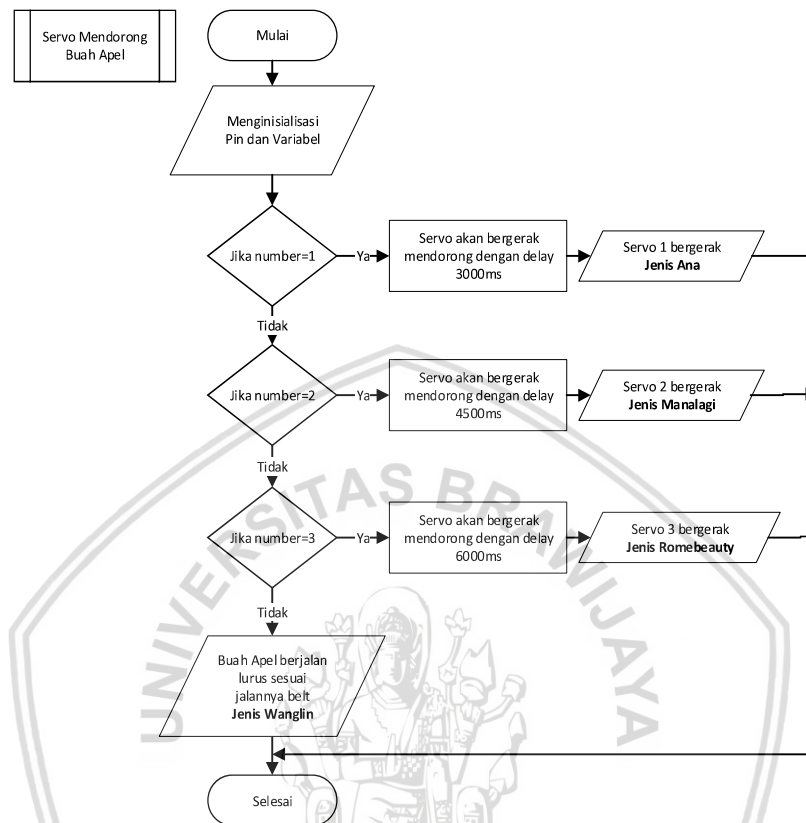


Gambar 5.9 Flowchart Klasifikasi

Berdasarkan Gambar 5.9 proses klasifikasi dilakukan setelah mendapatkan nilai dari data uji. Untuk mendapatkan nilai data uji tersebut dilakukan pengambilan 5 buah *sample* untuk masing-masing jenis sehingga didapat nilai-nilai *threshold* tersebut. Nilai-nilai tersebut akan muncul pada compiler program beserta nama apel yang terdeteksi. Selanjutnya diseleksi pada percabangan pertama, jika nilai berada di dalam batas semua maka masuk pada klasifikasi apel ana, jika tidak akan diseleksi lagi apakah masuk pada percabangan kedua, jika iya maka masuk klasifikasi apel manalagi. Proses berlanjut hingga percabangan

terakhir. jika nilai apel tidak berada pada keempat percabangan tersebut maka apel tidak masuk pada klasifikasi manapun.

5.1.3.3 Menjalankan Servo



Gambar 5.10 Flowchart Menjalankan Servo

Berdasarkan Gambar 5.10 dalam menjalankan servo dimulai dengan inisialisasi pin dan variabel yang akan digunakan. Setelah itu jika hasil dari klasifikasi memasuki jenis apel Ana servo 1 akan mendorong, jika jenis apel Manalagi servo 2 akan mendorong, jika jenis apel Romebeauty maka servo 3 akan mendorong, dan jika masuk jenis apel Wanglin maka apel akan berjalan lurus mengikuti jalannya konveyor.

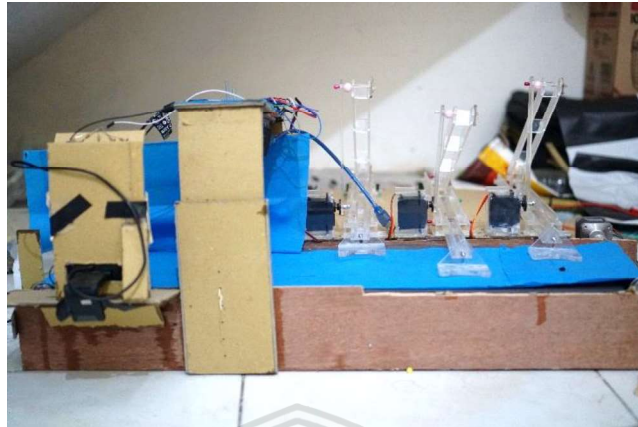
5.2 Implementasi Sistem

Dalam implementasi sistem ini, dilakukan pembuatan sistem yang direalisasikan dari perancangan sistem yang telah dilakukan sebelumnya pada subbab 5.1. Pada bagian implemmentasi sistem ini, akan dijelaskan secara rinci mengenai bagaimana mengimplementasikan prototype alat, kemudian implementasi dari perangkat keras, dan implementasi perangkat lunak.

5.2.1 Implementasi *Prototype* Alat Klasifikasi Jenis Apel

Dalam implementasi *prototype* mengacu pada Gambar 5.1 yang merupakan alat yaitu konveyor. Dalam pembuatan prototype alat menggunakan bahan triplek untuk badan, kain yang berfungsi layaknya *belt* konveyor, karton yang digunakan

sebagai penyanggah kamera dan akrilik dengan ketebalan 2mm yang digunakan sebagai lengan pendorong.

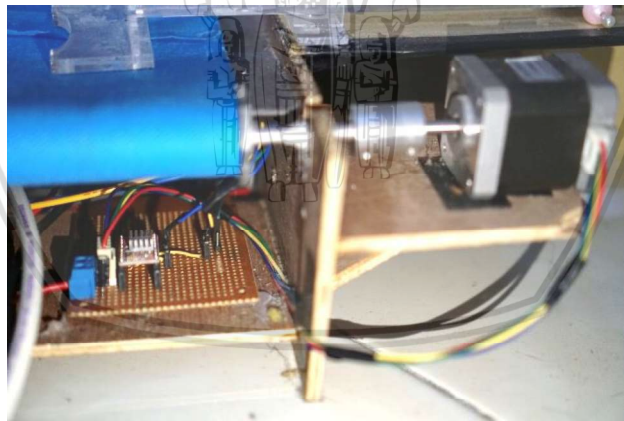


Gambar 5.11 Tampilan Alat Secara Keseluruhan

5.2.2 Implementasi Perangkat Keras

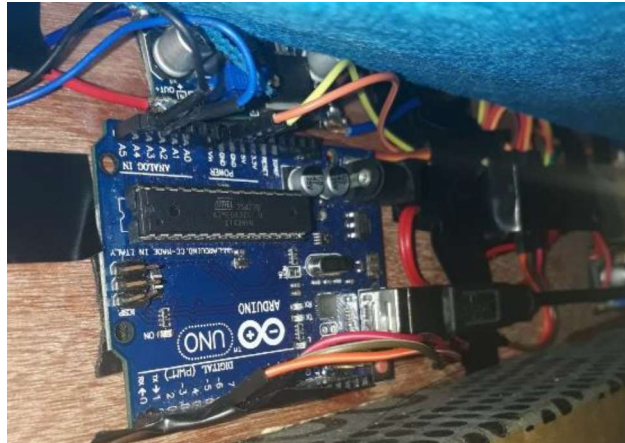
Pada tahap ini dilakukan penjelasan mengenai perancangan perangkat keras yang mencakup semua komponen yang digunakan dalam sistem ini. Seluruh komponen yang digunakan dirangkai menjadi satu lalu diletakkan pada bagian bawah *belt* agar terlihat rapi. Sesuai dengan subbab 5.1.2.

Pada Gambar 5.12 menghubungkan antara Motor *Stepper* dengan *driver* A4988. Motor driver ini dapat digunakan untuk mengendalikan Motor *Stepper*.



Gambar 5.12 Menghubungkan Motor Stepper dengan Driver

Gambar 5.13 menghubungkan driver dengan Arduino Uno. Penghubungan kedua komponen tersebut bertujuan untuk mengatur arah putar dan kecepatan putaran dari Motor *Stepper* yang diatur dengan *code* pada IDE Arduino.



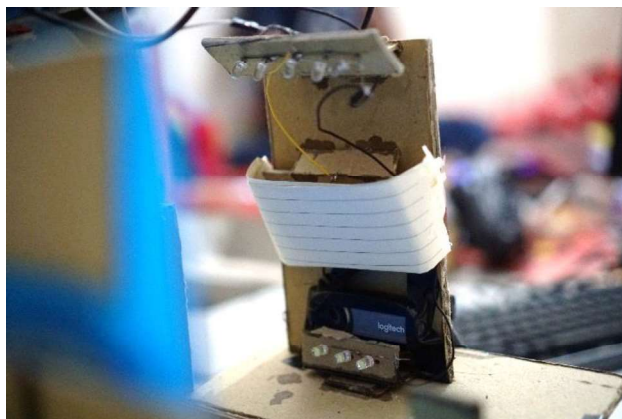
Gambar 5.13 Menghubungkan Driver dengan Arduino Uno

Pada Gambar 5.14 merupakan Raspberry Pi 3. Memiliki 4 port untuk dapat dihubungkan sesuai kebutuhan, tetapi pada sistem ini hanya digunakan 3 port yang masing-masing port untuk *webcam*, *mouse* dan *keyboard*.



Gambar 5.14 Menghubungkan Raspberry Pi 3 dengan Webcam Logitech

Gambar 5.15 merupakan tampilan webcam yang direkatkan pada karton guna membuat posisi webcam tetap sehingga pada saat dilakukan pengambilan gambar webcam tidak goyang.



Gambar 5.15 Tampilan Webcam Logitech

Pada Gambar 5.16 adalah menghubungkan Raspberry Pi 3 dengan Arduino Uno dengan menggunakan komunikasi serial I2C pada SDA dan SCL. Fungsi dari kedua pin tersebut adalah untuk saling mengirim dan menerima data.



Gambar 5.16 Menghubungkan Raspberry Pi 3 dengan Arduino Uno

Pada Gambar 5.17, guna menggerakkan lengan pendorong pada gambar sebelah kanan perlu menghubungkan tiap-tiap Motor Servo dengan Pin D3, D5 dan D6, lalu VCC dan GND dari ketiga Motor Servo dipararel pada Arduino Uno. Terlihat pada Gambar 5.17



Gambar 5.17 Menghubungkan Arduino Uno dengan Motor Servo

5.2.3 Implementasi Perangkat Lunak

Implementasi perangkat lunak pada klasifikasi buah apel berdasarkan subbab 5.1.3, dalam penulisan coding menggunakan *opencv* untuk melakukan proses pengolahan citra dan klasifikasi. Sedangkan untuk mengatur pergerakan Motor Stepper dan Motor Servo menggunakan Arduino IDE guna menggerakkan lengan pendorong untuk berjalan sesuai klasifikasinya. Pada Tabel 5.8 merupakan tabel yang digunakan untuk mengimplementasikan *library* yang digunakan. Penggunaan *library* “`matplotlib.pyplot as plt`” adalah *library plotting* 2D Python yang dapat digunakan untuk membuat plot. *Library* “`numpy as np`” digunakan untuk operasi matematika diantaranya operasi vector dan matriks. Penggunaan “`skimage.feature`” digunakan untuk menghitung GLCM dengan melakukan import pada “`greycomatrix`” sedangkan “`greycoprops`” digunakan untuk menghitung properti pada GLCM. Untuk memanggil *library* *cv2* digunakan “`import cv2`”. Untuk menghitung waktu komputasi digunakan “`import time`”. Untuk melakukan perhitungan matematika digunakan *library* “`import math`”. Dan yang terakhir *library* *SMBus* digunakan untuk komunikasi serial dengan menggunakan I2C.

Tabel 5.8 Inisialisasi Library Pada Raspberry Pi 3

Baris	Kode Program
1	<code>import matplotlib.pyplot as plt</code>
2	<code>import numpy as np</code>
3	<code>from skimage.feature import greycomatrix, greycoprops</code>
4	<code>from pylab import array, plot, show, axis, arange,</code>
5	<code>figure, uint8</code>
6	<code>import cv2</code>
7	<code>import time</code>
8	<code>import math</code>
9	<code>from smbus import SMBus</code>
	<code>...</code>

5.2.3.1 Implementasi Kode Program Motor Stepper

Pada implemtasi kode program motor *stepper* yang dilakukan pertama kali yaitu inisialisasi variabel “`unsigned long STEPPER_PULSE_HEIGH, STEPPER_PULSE_LOW; dan stop_stepper = true; number=0;`”. Setelah itu dilakukan inisialisasi pin yang digunakan yaitu pin 9 sebagai dir atau *direction* untuk menentukan *stepper* berjalan maju atau mundur, sedangkan pin 10 digunakan untuk step dari *stepper*. Pada baris 8 pin 9 diberi nilai *HEIGH* untuk menentukan arah pergerakan *stepper* maju. Pada baris 10-29 digunakan untuk menjalankan *stepper* secara terus dengan *delay* sebesar 2500 *microsecond*. Pada baris 26 ini digunakan untuk menghentikan *stepper* jika mendapat masukan berupa *number* 4 dari Raspberry Pi, sedangkan jika mendapat masukan *number*=5 maka *stepper* akan kembali berjalan.

Tabel 5.9 Kode Program Motor Stepper

Baris	Kode Program
1	unsigned long STEPPER_PULSE_HEIGH,
2	STEPPER_PULSE_LOW;
3	stop_stepper = true; number=0;
4	void setup() {
5	// put your setup code here, to run once:
6	pinMode(9, OUTPUT); //dir
7	pinMode(10, OUTPUT); //step
8	
9	digitalWrite(9, HEIGH); // arah maju
10	}
11	void loop() {
12	if (micros() >= STEPPER_PULSE_HEIGH){
13	digitalWrite(10, HEIGH);
14	STEPPER_PULSE_LOW = micros() + 2500;
15	STEPPER_PULSE_HEIGH = STEPPER_PULSE_LOW + 2500;
16	}
17	if (micros() >= STEPPER_PULSE_LOW){
18	digitalWrite(10, LOW);
19	}
20	}
21	void receiveData(int byteCount){
22	while(Wire.available()) {
23	number = Wire.read();
24	.
25	.
26	.
27	} else if (number == 4){
28	stop_stepper = true;
29	} else if (number == 5){
30	stop_stepper = false;
31	}
32	number = 0;
33	}
	}

5.2.3.2 Implementasi Kode Program Pengambilan Data Citra

Berikut adalah kode program yang digunakan untuk mengambil data citra, dimulai dengan inisialisasi *webcam* yang digunakan. Untuk membaca nilai *camera* digunakan perintah *cap.read*. *cap.read* mengembalikan nilai *bool* (*True/False*), jika *frame* dibaca dengan benar, maka akan bernilai *True*, sedangkan jika tidak akan bernilai *False*.

Tabel 5.10 Inisialisasi Webcam

Baris	Kode Program
1	cap = cv2.VideoCapture(1)
2	width = cap.get(3)
3	height = cap.get(4)
	...

Tabel 5.11 Membaca Nilai Webcam

Baris	Kode Program
1	while(True):
2	ret, frame = cap.read()
3	
4	if ret == False:
5	break
	...

Tabel 5.12 Pengolahan Citra Digital

Baris	Kode Program
1	newFrame =
2	(maxIntensity/phi)*(frame/(maxIntensity/theta))*10
3	newFrame = array(newFrame, dtype=uint8)
4	
5	ycbcr = cv2.cvtColor(frame, cv2.COLOR_BGR2YCrCb)
6	
7	colormask_ycryb = cv2.inRange(ycbcr, bottomLimit,
8	topLimit)
9	
10	opening = cv2.morphologyEx(colormask_ycryb,
11	cv2.MORPH_OPEN,
12	cv2.getStructuringElement(cv2.MORPH_RECT,(3,3)),
13	iterations = 6)
14	
15	_, contours, _ = cv2.findContours(opening,
16	cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
17	bitwise = cv2.bitwise_and(newFrame, newFrame, mask
18	= opening)
19	...

Pada tahap pengolahan citra digital yaitu dilakukan perubahan format warna dari RGB ke YcbCr yang digunakan untuk memisahkan *background* dan *foreground*. Setelah itu dilakukan proses *Morphology* yaitu *opening* yang berfungsi untuk memperbaiki hasil segmentasi citra. Proses *opening* dimulai dengan erosi yaitu untuk memperkecil tepi objek. Setelah itu dilakukan proses dilasi yaitu memperbesar segmen citra dengan menambah lapisan di sekitar objek. Proses *opening* dilakukan untuk menghilangkan objek-objek kecil untuk menjadikan tepian citra menjadi halus agar citra inputan menjadi bentuk yang diinginkan. Setelah itu dilakukan proses *contour* untuk menentukan area objek yang akan dideteksi.

Tabel 5.13 Pendeteksian Objek

Baris	Kode Program
1	#membaca objek, lalu diberi kotak hijau
2	for cnt in contours:
3	area = cv2.contourArea(cnt)
4	if area > 1500:

```

5         status = 1
6         x, y, w, h = cv2.boundingRect(cnt)
7     #centroid
8         center = (int(x + (w/2)), int(y + (h/2)))
9     #membuat 3 kotak untuk sample glcm
10        pos_colorcrop = [((center[0]-
11 40),(center[1]-10)),((center[0]-20), (center[1]+10))]
12        pos_samples = [
13            [((center[0]-35),(center[1]-
14 35)),((center[0]-5), (center[1]-5))],
15            [((center[0]+10),(center[1]-
16 25)),((center[0]+40), (center[1]+5))],
17            [((center[0]-
18 5),(center[1]+10)),((center[0]+25), (center[1]+40))]
19        ]
20
21    cv2.rectangle(frame,(x,y),(x+w,y+h),(0,255,0),1)
22        cv2.circle(frame, center, 3, (0, 255, 0),
23 -1)
24
25    cv2.rectangle(frame,pos_colorcrop[0],pos_colorcrop[1
26 ],(0,255,0),1)
27    for pos_sample in pos_samples:
28
29    cv2.rectangle(frame,pos_sample[0],pos_sample[1],(0,2
30 55,0),1)
31        if (center[0] > (int(320/2)-10)) and
32 (center[0] < (int(320/2)+10)):
33
34            bus.write_byte(0x24,4)
35
36            detecting = True
37
38            if detecting:
39
40                counter += 1
41
42                if counter == 35:
43
44                    status = 2
45
46    #aspectratio
47        aspectratio = 1.00
48        if w != 0 and h != 0:
49            aspectratio = float(w) /
50 float (h) * 100
51
52        sarraster = cv2.cvtColor(frame2,
53 cv2.COLOR_BGR2GRAY)
54
55        sampel = []
56
57        for pos_sample in pos_samples:

```

```

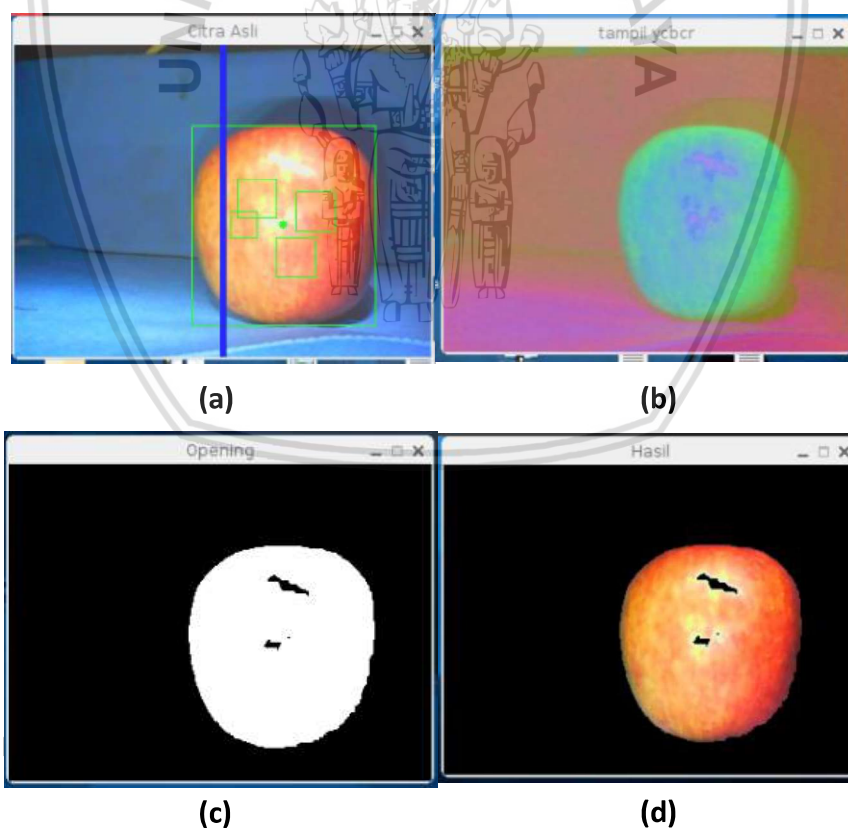
58
59 sampel.append(sarraster[pos_sample[0][1]:pos_sample[
60 1][1],pos_sample[0][0]:pos_sample[1][0]])
61         colorcropped =
62 frame2[pos_colorcrop[0][1]:pos_colorcrop[1][1],
63 pos_colorcrop[0][0]:pos_colorcrop[1][0]]
64         colorcropped_hsv =
65 cv2.cvtColor(colorcropped, cv2.COLOR_BGR2HSV)
66
67         contrastraster = np.copy(sampel)
68         contrastraster[:] = 0
69
70         for a in range(len(sampel)) :
71 #glcm
72             for i in
73 range(sampel[a].shape[0]):
74                 for j in
75 range(sampel[a].shape[1] ):
76                     if (i-4) < 0 or (i+4)
77 > sampel[a].shape[0] or (j-4) < 0 or (j+4) >
78 sampel[a].shape[1]:
79                         continue
80
81                         glcm_window =
82 sampel[a][i-4: i+4, j-4 : j+4]
83
84                         glcm =
85 greycomatrix(glcm_window, [1], [0], 256, symmetric =
86 True, normed = True )
87
88
89         contrastraster[a][i,j] =         greycoprops(glcm,
90 'contrast')
91
92                         glcm = None
93                         glcm_window = None
94
95         for b in range(len(sampel)):
96             s = cv2.resize(sampel[b],
97 (0,0), fx=3, fy=3)
98
99         cv2.imshow("contrast"+str(b), s)
100
101             glcm_contrast =
102 np.mean(contrastraster[contrastraster>np.max(contras
103 traster)/2])
104             h = np.mean(colorcropped[:, :, 0])
105         ...

```

Pada tahap pendeteksian objek ini dilakukan pembacaan *countour* pada objek yang diberi *bounding rectangle*. Jika objek yang dibaca memiliki piksel lebih dari 1500 maka objek akan diberi kotak yang tertulis pada baris 2-6. Lalu ditentukan titik tengah atau *centroid* seperti pada baris 8. Pada baris 10-18 dibuat kotak-kotak

kecil berukuran 30x30 piksel sebanyak 3 buah untuk mengambil *sample* GLCM. *Bounding rect* ini diberi warna hijau, terlihat dengan diberi nilai 255 pada warna *Green*, di tengah-tengah objek juga terdapat titik kecil yang juga diberi warna hijau, dengan diberi nilai 255 tertulis pada baris 21-23. Pada *frame* diberi garis yang posisinya ditengah untuk menentukan bahwa pada saat posisi tersebut objek akan terdeteksi seperti pada baris 31-32. Pada baris 46-49 untuk *aspectratio* yang berfungsi untuk menentukan ukuran objek. Dengan melakukan perhitungan tinggi (*Heigh*) dan lebar (*width*) piksel. Setelah itu objek akan diganti format warna menjadi keabu-abuan.

Setelah itu akan diganti format warna menjadi HSV, disini hanya mengambil nilai *Hue* saja. Pada penelitian ini hanya menggunakan nilai *Hue* dikarenakan *Hue* dapat menyatakan warna yang sebenarnya dan untuk menentukan warna kehijauan (*greeness*). Pada GLCM terdapat beberapa fitur yang dapat digunakan, salah satunya yaitu *contras*. Pada penelitian ini hanya menggunakan fitur tersebut dikarenakan fitur tersebut dapat menampilkan hasil sesuai dengan inputan, sedangkan fitur lain seperti *dissimilarity*, *entropy* tidak dapat menampilkan hasil. GLCM digunakan untuk mendeteksi tekstur dari salah satu buah apel yaitu Apel Wanglin. GLCM bekerja dengan melihat piksel pertama dari objek, lalu mendeteksi piksel tetangganya, dengan melihat 4 piksel tetangga kiri, kanan, atas, dan bawah. Proses tersebut diulang hingga piksel terakhir seperti pada baris 71-96.

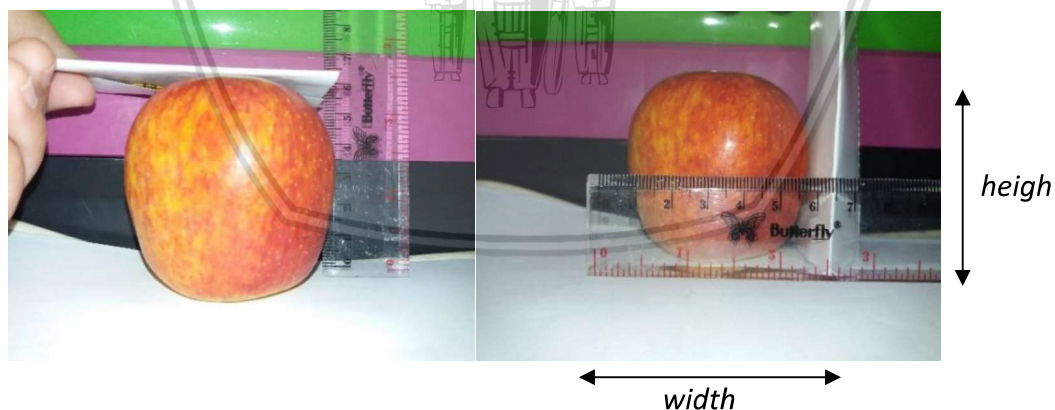


Gambar 5.18 (a) Citra yang sudah di bounding box, (b) Citra dikonversi ke YCbCr, (c) Hasil Opening, (d) Hasil opening digabungkan dengan citra asli

Pada Gambar 5.18 (a) citra yang ditangkap oleh *webcam* berupa gambar buah apel. Pada *frame* dibuat garis tengah yang berfungsi untuk mendeteksi objek. Jika objek terkena garis tersebut maka sistem akan mendeteksi adanya sebuah objek. Terdapat 3 buah kotak kecil berukuran 30x30 yang digunakan untuk mengambil *sample* untuk proses GLCM dan 1 kotak yang digunakan untuk mengambil *sample* warna *Hue*. Lalu hasil dari citra masukan dikonversi ke dari RGB ke YCbCr yang ditunjukkan oleh gambar (b). Pada gambar (c) hasil dari YCbCr di *opening* yang bertujuan untuk memperbaiki hasil segmentasi citra. Proses *opening* dimulai dengan erosi yaitu untuk memperkecil tepi objek. Setelah itu dilakukan proses dilasi yaitu memperbesar segmen citra dengan menambah lapisan di sekitar objek. Proses *opening* dilakukan untuk menghilangkan objek-objek kecil untuk menjadikan tepian citra menjadi halus agar citra masukan menjadi bentuk yang diinginkan. Gambar (d) hasil dari *opening* di gabungkan dengan gambar asli yang sudah dihilangkan *background* untuk menampilkan warna asli buah apel.

Pada baris 46 Tabel 5.12 dilakukan proses perhitungan *aspectratio* untuk menentukan ukuran dari buah apel. Perhitungan *aspectratio* ini menggunakan nilai tinggi (*Heigh*) dan lebar (*width*) dari objek yang sudah di *bounding box*. Dibawah ini dilakukan perhitungan untuk 2 jenis apel, yaitu apel ana dan manalagi. Diambil dua contoh saja dikarenakan apel wanglin dan romebeauty memiliki bentuk yang sama seperti apel manalagi.

Pengambilan gambar dilakukan dengan menggunakan kamera *smartphone*. Untuk mendapatkan nilai lebar dan tinggi. Buah apel diletakkan pada bidang yang datar, lalu diukur dengan menggunakan penggaris dengan satuan *centimeter* pada masing-masing lebar dan tinggi. Berdasarkan **Persamaan 2.12** dapat dihitung nilai *aspectratio* :

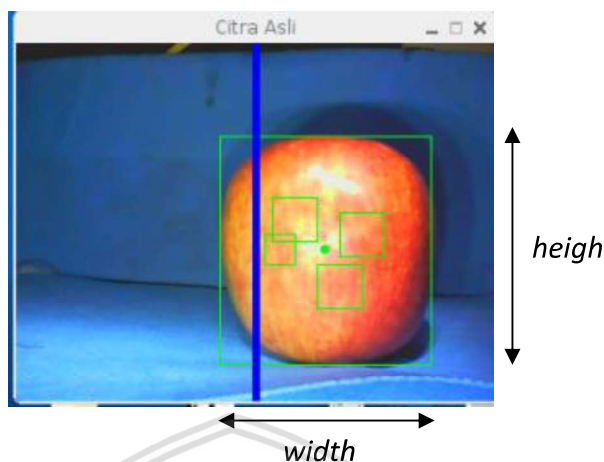


Gambar 5.19 Pengukuran manual dengan menggunakan penggaris

Adapun rumus yang digunakan adalah sebagai berikut :

$$\begin{aligned} \text{Aspectratio} &= \frac{\text{width}}{\text{high}} \times 100\% \\ &= \frac{6.5}{5.5} \times 100\% = 1.1818 \times 100 = 118.181 \end{aligned}$$

Sedangkankan untuk mendapatkan nilai tinggi dan lebar dari hasil percobaan adalah sebagai berikut :



Gambar 5.20 Gambar buah apel yang ditangkap oleh webcam

width : 141
high : 152

Gambar 5.21 Nilai lebar dan tinggi

Dari tersebut dilakukan perhitungan sebagai berikut :

$$\begin{aligned} \text{Aspectratio} &= \frac{\text{width}}{\text{heigh}} \times 100\% \\ &= \frac{141}{152} \times 100\% = 0.927 \times 100 = 92.763 \end{aligned}$$

Untuk menunjukkan kebenaran dari perhitungan manual, dapat dari hasil percobaan adalah 92.763. Sedangkan hasil perhitungan manual adalah 1.181. Terdapat selisih antara perhitungan manual dengan hasil percobaan ini dikarenakan hanya menggunakan penggaris yang tidak dapat ditentukan nilai *millimeter*. Adapun selisih dari perhitungan manual dengan hasil percobaan dengan menggunakan rumus sebagai berikut :

$$\begin{aligned} \text{Selisih nilai} &= \text{perhitungan manual} - \text{hasil percobaan} \\ &= 118.181 - 92.763 = 25.418 \end{aligned}$$

Dari hasil selisih nilai diatas, dapat dilakukan proses perhitungan *presentase error* dengan rumus sebagai berikut:

$$\begin{aligned} \text{Presentase error} &= \frac{\text{Selisih Nilai}}{\text{Perhitungan manual}} \times 100\% \\ &= \frac{25.418}{118.181} \times 100\% \\ &= 21.5 \% \end{aligned}$$

Sedangkan untuk apel manalagi dengan perhitungan sebagai berikut :

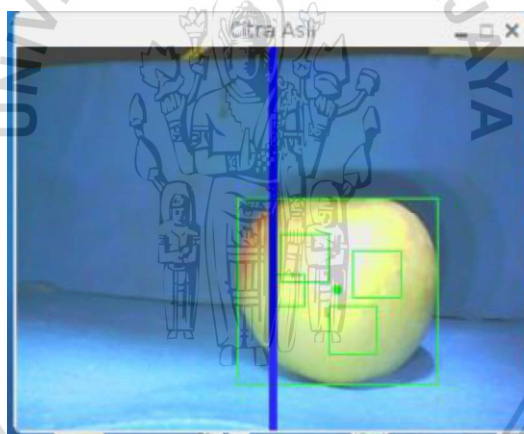


Gambar 5.22 Pengukuran manual dengan menggunakan penggaris

Adapun rumus yang digunakan adalah sebagai berikut :

$$\begin{aligned} \text{Aspectratio} &= \frac{\text{width}}{\text{heigh}} \times 100\% \\ &= \frac{4.8}{5.5} \times 100\% = 1.0683 \times 100 = 106.837 \end{aligned}$$

Sedangkankan untuk mendapatkan nilai tinggi dan lebar dari hasil percobaan adalah sebagai berikut :



Gambar 5.23 Gambar buah apel yang ditangkap oleh webcam

width : 125
high : 117

Gambar 5.24 Nilai lebar dan tinggi

$$\begin{aligned} \text{Aspectratio} &= \frac{\text{width}}{\text{heigh}} \times 100\% \\ &= \frac{125}{117} \times 100\% = 1.145 \times 100 = 114.545 \end{aligned}$$

Untuk menunjukkan kebenaran dari perhitungan manual, dapat dari hasil percobaan adalah 114.545. Sedangkan hasil perhitungan manual adalah 106.837. Terdapat selisih antara perhitungan manual dengan hasil percobaan ini dikarenakan hanya menggunakan penggaris yang tidak dapat ditentukan nilai

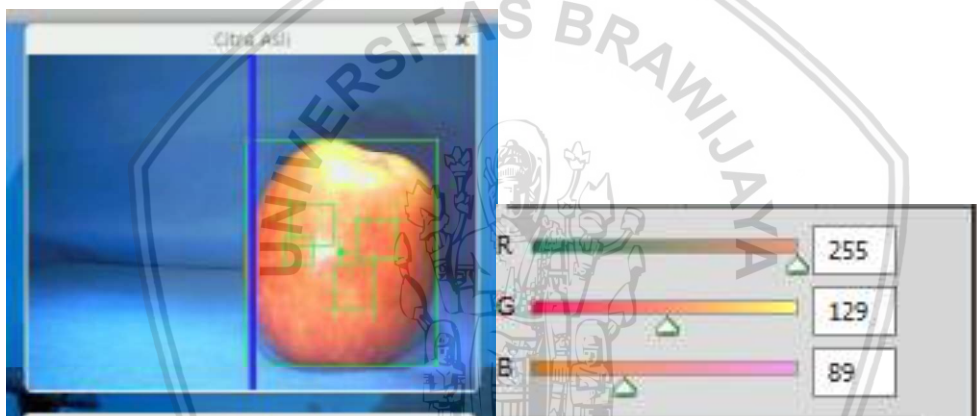
millimeter. Adapun selisih dari perhitungan manual dengan hasil percobaan dengan menggunakan rumus sebagai berikut :

$$\begin{aligned}\text{Selisih nilai} &= | \text{perhitungan manual} - \text{hasil percobaan} | \\ &= | 106.683 - 114.545 | = 7.862\end{aligned}$$

Dari hasil selisih nilai diatas, dapat dilakukan proses perhitungan *presentase error* dengan rumus sebagai berikut:

$$\begin{aligned}\text{Presentase error} &= \frac{\text{Selisih Nilai}}{\text{Perhitungan manual}} \times 100\% \\ &= \frac{7.862}{106.683} \times 100\% \\ &= 7.369 \%\end{aligned}$$

Sedangkan untuk menghitung nilai *Hue* pada HSV dapat dihitung dengan **Persamaan (2.2)** dan **Persamaan (2.3)**.



Gambar 5.25 Nilai RGB

$$R : 225 / 255 = 1$$

$$G : 129 / 255 = 0.505$$

$$B : 89 / 255 = 0.349$$

$$C_{\text{Max}} = R : 1$$

$$C_{\text{Min}} = B : 0.349$$

$$\Delta = 1 - 0.349 = 0.651$$

Hue Calculation

$$\begin{aligned}H &= 60^\circ \times \left(\frac{G' - B'}{\Delta} \text{ mod } 6 \right) \\ &= 60^\circ \times \left(\frac{0.505 - 0.349}{0.651} \text{ mod } 6 \right) \\ &= 60^\circ \times \left(\frac{0.156}{0.651} \text{ mod } 6 \right) \\ &= 60^\circ \times (0.239 \text{ mod } 6) = 14.34\end{aligned}$$

Tabel 5.14 Proses Klasifikasi

Baris	Kode Program
1	if (hh >= 60 and hh <= 123) and (aspectratio <=96) and
2	(glcm_contrast >= 24 and glcm_contrast <= 179):
3	print ("Apel Ana")
4	bus.write_byte(0x24, 1)
5	if (hh >= 84 and hh <= 235) and
6	(aspectratio >= 97 and aspectratio <= 121) and
7	(glcm_contrast >= 8 and glcm_contrast <= 40):
8	print ("Apel Manalagi")
9	bus.write_byte(0x24, 2)
10	if (hh >= 62 and hh <= 137) and
11	(aspectratio >=106 and aspectratio <= 123) and
12	(glcm_contrast >= 73):
13	print ("Apel Rome Beauty")
14	bus.write_byte(0x24, 3)
15	if (hh >= 100 and hh <= 134) and
16	(aspectratio >= 100) and (glcm_contrast >= 9 and
17	glcm_contrast <= 55):
18	print ("Apel Wanglin")

Pada proses klasifikasi dengan menggunakan percabangan *if*. Dengan memproses nilai *Hue*, *aspectratio* dan GLCM *Contrast* yang di dapat dari hasil pendeteksian lalu nilai-nilai tersebut masuk pada jenis mana. Jika masuk pada *range* jenis apel Ana maka servo akan mendorong dengan menjalankan fungsi "bus.write_byte(0x24, 1)". Jika masuk pada *range* apel manalagi akan menjalankan fungsi pada "bus.write_byte(0x24, 2)". Jika masuk pada *range* apel romebeauty akan menjalankan fungsi "bus.write_byte(0x24, 3)". Jika wanglin maka akan berjalan lurus.

5.2.3.3 Implementasi Kode Program untuk Motor Servo

Dimulai dengan inisialisasi *library* yang digunakan, yaitu *library wire.h* yang merupakan protocol serial sinkronisasi I2C. lalu inisialisasi servo yang digunakan. Setelah itu ketiga servo diberi tipe data *bool false*. *Wire.onReceive(receiveData)* *sintax* ini digunakan untuk inisialisai alamat interface I2C dan ketika menerima data. Lalu pada baris 14-16 digunakan untuk inisialisasi pin pada Arduino. Pada baris ke 18 -20 digunakan untuk menentukan sudut awal saat yaitu 90°. Pada saat kulit selesai terdeteksi dan mendapatkan hasil, jika hasil adalah jenis apel ana, maka servo pertama akan bergerak dengan sudut sebesar 180°. Setelah mendorong apel lalu kondisi servo akan ebrgerak kembali ke posisis awal yaitu 90°. Proses tersebut berlanjut hingga baris 78. Pada baris 86 hingga 103 digunakan untuk menentukan waktu *milis* kapan servo akan bergerak pada saat *webcam* selesai mendeteksi. Pada saat servo pertama bergerak setelah waktu 3000s dari *webcam* setelah mendeteksi objek. Servo kedua bergerak setelah waktu 4500s dari *webcam*. Dan servo ketiga bergerak setelah waktu 6000s dari *webcam*.

Tabel 5.15 Kode Program Motor Servo

Baris	Kode Program
1	#include <Wire.h>
2	#include <Servo.h>
3	#define SLAVE_ADDR 0x24
4	Servo myservo1;
5	Servo myservo2;
6	Servo myservo3
7	bool servol = false, servo2 = false, servo3 = false,
8	int number = 0;
9	bool pulse_state = false;
10	void setup() {
11	Wire.begin(SLAVE_ADDR);
12	Wire.onReceive(receiveData);
13	myservo1.attach(3);
14	myservo2.attach(5);
15	myservo3.attach(6);
16	
17	myservo1.write(90);
18	myservo2.write(80);
19	myservo3.write(80);
20	
21	Serial.begin(9600);
22	
23	pinMode(9, OUTPUT); //dir
24	pinMode(10, OUTPUT); //step
25	digitalWrite(9, HIGH); // arah maju
26	}
27	void loop() {
28	if (servol == true){
29	if (millis() >= WAIT_A_DOWN){
30	// Serial.println("Servo 1 Down");
31	myservo1.write(180);
32	}
33	if (millis() >= WAIT_A_UP){
34	myservo1.write(90);
35	servol = false;
36	// Serial.println("Servo 1 Up");
37	}
38	}
39	if (servo2 == true){
40	if (millis() >= WAIT_B_DOWN){
41	// Serial.println("Servo 2 Down");
42	myservo2.write(170);
43	}
44	if (millis() >= WAIT_B_UP){
45	myservo2.write(80);
46	servo2 = false;
47	// Serial.println("Servo 2 Up");
48	}
49	}
50	if (servo3 == true){

```

51     if (millis() >= WAIT_C_DOWN){
52         //      Serial.println("Servo 3 Down");
53         myservo3.write(170);
54     }
55     if (millis() >= WAIT_C_UP){
56         myservo3.write(80);
57         servo3 = false;
58         //      Serial.println("Servo 3 Up");
59     }
60 }
61 }
62 void receiveData(int byteCount){
63     while(Wire.available()) {
64         number = Wire.read();
65
66         if (number == 1){
67             servol = true;
68             WAIT_A_DOWN = millis() + 3000;
69             WAIT_A_UP = WAIT_A_DOWN + 1000;
70         } else if (number == 2){
71             servo2 = true;
72             WAIT_B_DOWN = millis() + 4500;
73             WAIT_B_UP = WAIT_B_DOWN + 1000;
74         } else if (number == 3){
75             servo3 = true;
76             WAIT_C_DOWN = millis() + 6000;
77             WAIT_C_UP = WAIT_C_DOWN + 1000;
78         } else if (number == 4){
79             stop_stepper = true;
80         } else if (number == 5){
81             stop_stepper = false;
82         }
83         number = 0;
84     }
85 }

```

Dibawah ini adalah perhitungan manual pada saat mendorong servo yang mempunyai jarak dari kamera adalah servo 1 adalah 24 cm ,servo 2 adalah 35.8 cm dan servo 3 adalah 47.8. Mulanya 1 step pada stepper membutuhkan waktu $2500 \times 2 = 5000$ microsecond yang berarti 5 ms, 1 revolusi stepper mempunyai 200 step maka untuk menghitung kecepatan belt stepper adalah

$$\text{Kecepatan Stepper} = \frac{\text{jumlah 1 revolusi stepper} \cdot \text{waktu 1 step}}{1000}$$

$$\text{Kecepatan Stepper} = \frac{200 \cdot 5}{1000} = 1 \text{ rpd (round per detik)}$$

Kemudian keliling lingkaran pada roller belt konveyor adalah :

$$\text{Kel roller} = 2\pi \cdot r = \pi \cdot d$$

$$\text{Kel roller} = 3,14 \times 2,54 = 7,97 \text{ cm}$$

Untuk menghitung jarak antara area pengambilan data setelah dikomputasi dengan servo adalah adalah :

$$\text{Servo 1} = 24 \times 7,97 = 3 \text{ s atau } 3000 \text{ ms}$$

$$\text{Servo 2} = 35.8 \times 7,97 = 4.5 \text{ s atau } 4500 \text{ ms}$$

$$\text{Servo 3} = 47.8 \times 7,97 = 6 \text{ s atau } 6000 \text{ ms}$$

Jadi saat mendorong apel sesuai dengan kelasnya membutuhkan *delay* masing-masing sebanyak 3000 ms, 4500 ms dan 6000 ms.



BAB 6 PENGUJIAN DAN ANALISIS

6.1 Pengujian Akurasi Nilai *Aspectratio*

6.1.1 Tujuan Pengujian

Pengujian ini dilakukan untuk menentukan keakuratan nilai *aspectratio* dari keempat jenis apel dengan melihat hasil perbandingan antara tinggi dan lebar.

6.1.2 Prosedur Pengujian

Prosedur pengujian dilakukan agar pengujian berjalan dengan semestinya, adapun langkah-langkah yang dilakukan adalah sebagai berikut :

1. Menghubungkan *powersupply* ke sumber tegangan untuk menyalakan seluruh komponen.
2. Upload program ke Arduino Uno.
3. Menjalankan program pada opencv untuk pengolahan citra.
4. Meletakkan objek pada konveyor pada jarak yang sudah ditentukan dari posisi webcam.
5. Hasil dari nilai *aspectratio* akan muncul pada *compiler* pada aplikasi opencv. Dari nilai tersebut selanjutnya dilakukan perhitungan akurasi dengan menggunakan rumus :

$$Akurasi = \frac{Total\ data - Data\ tidak\ sesuai}{Total\ Data} \times 100\%$$

6.1.3 Pelaksanaan Pengujian

Pelaksanaan pengujian dilakukan sesuai dengan prosedur yang sudah dilakukan pada subbab sebelumnya. pengujian ini dilakukan sebanyak 10 kali untuk masing-masing jenis apel, sehingga terdapat 40 kali pengujian yang dilakukan.

6.1.4 Hasil dan Analisis Pengujian

Tabel hasil pengujian akurasi nilai *aspectratio*, hasil ditunjukkan pada tabel dibawah ini.

Tabel 6.1 Pengujian Nilai Aspectratio Apel Ana

Uji Ke-	Aspectratio	Hasil Klasifikasi	Apel Sebenarnya	Kesesuaian
1	88.965	Ana	Ana	Sesuai
2	89.361	Ana	Ana	Sesuai
3	80.392	Ana	Ana	Sesuai
4	87.671	Ana	Ana	Sesuai
5	88.356	Ana	Ana	Sesuai

6	77.272	Ana	Ana	Sesuai
7	90.344	Ana	Ana	Sesuai
8	83.544	Ana	Ana	Sesuai
9	83.125	Ana	Ana	Sesuai
10	99.23	Tidak Terdeteksi	Ana	Tidak Sesuai
Hasil Akurasi				90 %

Berdasarkan hasil pengujian tersebut terdapat 1 buah apel yang tidak terdeteksi dikarenakan nilai *aspectratio* lebih dari *range* pengujian yang dilakukan.

Untuk menghitung akurasi kesesuaian dari *aspectratio* adalah sebagai berikut :

$$Akurasi = \frac{Total\ data - Data\ tidak\ sesuai}{Total\ Data} \times 100\%$$

$$Akurasi = \frac{10 - 1}{10} \times 100\%$$

$$= 90 \%$$

Tabel 6.2 Pengujian Aspectratio Apel Manalagi

Uji Ke-	Aspectratio	Hasil Klasifikasi	Apel Sebenarnya	Kesesuaian
1	120.472	Manalagi	Manalagi	Sesuai
2	109.459	Manalagi	Manalagi	Sesuai
3	109.836	Manalagi	Manalagi	Sesuai
4	109.929	Manalagi	Manalagi	Sesuai
5	103.246	Manalagi	Manalagi	Sesuai
6	106.122	Manalagi	Manalagi	Sesuai
7	111.25	Manalagi	Manalagi	Sesuai
8	108.196	Manalagi	Manalagi	Sesuai
9	108.196	Manalagi	Manalagi	Sesuai
10	106.837	Tidak Terdeteksi	Manalagi	Tidak Sesuai
Hasil Akurasi				90%

Berdasarkan hasil pengujian tersebut terdapat 1 buah apel yang tidak terdeteksi, dikarenakan nilai dari hasil pengujian melebihi *range* salah satu komponen yaitu GLCM.

Untuk menghitung akurasi kesesuaian dari *aspectratio* adalah sebagai berikut :

$$Akurasi = \frac{Total\ data - Data\ tidak\ sesuai}{Total\ Data} \times 100\%$$

$$Akurasi = \frac{10 - 1}{10} \times 100\%$$

$$= 90 \%$$

Tabel 6.3 Pengujian Aspectratio Apel Romebeauty

Uji Ke-	Aspectratio	Hasil Klasifikasi	Apel Sebenarnya	Kesesuaian
1	122.222	Romebeauty	Romebeauty	Sesuai
2	116.981	Romebeauty	Romebeauty	Sesuai
3	117.592	Romebeauty	Romebeauty	Sesuai
4	118.518	Romebeauty	Romebeauty	Sesuai
5	116.666	Romebeauty	Romebeauty	Sesuai
6	120.754	Romebeauty	Romebeauty	Sesuai
7	123.931	Tidak Terdeteksi	Romebeauty	Tidak Sesuai
8	114.285	Tidak Terdeteksi	Romebeauty	Tidak Sesuai
9	118.095	Tidak Terdeteksi	Romebeauty	Tidak Sesuai
10	121.428	Tidak Terdeteksi	Romebeauty	Tidak Sesuai
Hasil Akurasi				60%

Dari hasil pengujian apel romebeauty terdapat 4 jenis apel yang tidak terdeteksi, dikarenakan terdapat bintik yang kontras dengan warna kulit yaitu merah sehingga dianggap memiliki tekstur yang banyak dan juga warna merah yang dimiliki apel yang tidak terdeteksi terlalu kecil yang sama dengan nilai dari apel ana. Sehingga nilai dari hasil pengujian melebihi *range* yang sudah ditentukan.

Untuk menghitung akurasi kesesuaian dari *aspectratio* adalah sebagai berikut :

$$Akurasi = \frac{Total\ data - Data\ tidak\ sesuai}{Total\ Data} \times 100\%$$

$$Akurasi = \frac{10 - 4}{10} \times 100\%$$

$$= 60 \%$$

Tabel 6.4 Pengujian Aspectratio Apel Wanglin

Uji Ke-	Aspectratio	Hasil Klasifikasi	Apel Sebenarnya	Kesesuaian
1	107.575	Wanglin	Wanglin	Sesuai
2	107.938	Wanglin	Wanglin	Sesuai
3	107.518	Wanglin	Wanglin	Sesuai
4	109.09	Wanglin	Wanglin	Sesuai
5	106.06	Wanglin	Wanglin	Sesuai
6	102.721	Wanglin	Wanglin	Sesuai
7	102.739	Wanglin	Wanglin	Sesuai
8	104.379	Wanglin	Wanglin	Sesuai
9	108.73	Tidak Terdeteksi	Wanglin	Tidak Sesuai
10	104.697	Tidak Terdeteksi	Wanglin	Tidak Sesuai
Hasil Akurasi				80%

Berdasarkan hasil pengujian yang dilakukan untuk apel wanglin, terdapat dua buah apel yang tidak terdeteksi, hal ini dikarenakan bentuk dan warna yang mirip dengan apel manalagi sehingga dianggap sebagai apel manalagi. Dan juga warna bintik yang hampr sama dengan kulit buah yang tidak terdeteksi oleh webcam.

Untuk menghitung akurasi kesesuaian dari *aspectratio* adalah sebagai berikut :

$$Akurasi = \frac{Total\ data - Data\ tidak\ sesuai}{Total\ Data} \times 100\%$$

$$Akurasi = \frac{10 - 2}{10} \times 100\%$$

$$= 80\%$$

Berdasarkan hasil pengujian pada Tabel 6.1 sampai Tabel 6.4 diatas telah dilakukan pengujian untuk apel ana, manalagi, romebeauty dan wanglin sebanyak 40 kali.

Tabel 6.5 Rata-rata akurasi *aspectratio*

No	Nama	Rata-rata <i>aspectratio</i>
1	Apel ana	90%
2	Apel manalagi	90%
3	Apel romebeauty	60%
4	Apel wanglin	80%
Rata-rata		80%

Dari Tabel 6.5 didapat hasil akurasi dari pengujian keempat jenis tersebut sebesar 80%.

Berikut disertakan salah satu hasil *aspectratio* pada *compiler*.

```
Hue : 108.0425
Aspect Ratio : 81.3559322034
GLCM Contrast : 142.90217391304347
-----
Apel Ana
```

Gambar 6.1 Hasil Pengujian *Aspectratio*

6.2 Pengujian Waktu Komputasi

6.2.1 Tujuan Pengujian

Tujuan dari dilakukannya pengujian ini adalah untuk mengetahui berapa lama waktu yang dibutuhkan oleh sistem untuk menyelesaikan proses dimulai dari webcam membaca citra objek hingga mendapatkan hasil klasifikasi.

6.2.2 Prosedur Pengujian

Prosedur pengujian dilakukan agar pengujian berjalan dengan semestinya, adapun langkah-langkah yang dilakukan adalah sebagai berikut :

1. Menghubungkan *powersupply* ke sumber tegangan untuk menyalakan seluruh komponen.
2. Upload program ke Arduino Uno.
3. Menjalankan program pada opencv untuk pengolahan citra.
4. Meletakkan objek pada konveyor pada jarak yang sudah ditentukan dari posisi webcam.
5. Setelah mendapat hasil waktu komputasi untuk masing-masing apel, maka langkah selanjutnya adalah menghitung rata-rata waktu komputasi.

6.2.3 Pelaksanaan Pengujian

Pelaksanaan pengujian dilakukan dengan melihat waktu komputasi dari 40 objek yang diujikan.

6.2.4 Hasil dan Analisis Pengujian

Berikut adalah hasil pengujian waktu komputasi yang dilakukan terhadap 40 buah apel.

Tabel 6.6 Pengujian Waktu Komputasi Apel Ana

Uji Ke-	Waktu Komputasi (ms)
1	15509
2	15784
3	15616
4	15601
5	16010
6	16075
7	15546
8	15819
9	15550
10	15625
Total	157135 ms
Rata-rata	15713.5 ms

Dari 10 kali percobaan yang dilakukan terhadap 10 apel ana yang berbeda, didapat rata-rata waktu komputasi sebesar 15713.5 ms atau sama dengan 15 detik.

Tabel 6.7 Pengujian Waktu Komputasi Apel Manalagi

Uji Ke-	Waktu Komputasi (ms)
1	16012
2	16257
3	16545
4	16026
5	16099
6	15926
7	15992
8	15742
9	15742
10	15464
Total	159805 ms
Rata-rata	15980.5 ms

Dari 10 kali percobaan yang dilakukan terhadap 10 apel manalagi yang berbeda, didapat rata-rata waktu komputasi sebesar 15980.5 ms atau sama dengan 15 detik.

Tabel 6.8 Pengujian Waktu Komputasi Apel Romebeauty

Uji Ke-	Waktu Komputasi (ms)
1	16683
2	15922
3	15706
4	16481
5	15682
6	15805
7	16298
8	15947
9	15673
10	15571
Total	159768 ms
Rata-rata	15976.8 ms

Dari 10 kali percobaan yang dilakukan terhadap 10 apel romebeauty yang berbeda, didapat rata-rata waktu komputasi sebesar 15976.8 ms atau sama dengan 15 detik

Tabel 6.9 Pengujian Waktu Komputasi Apel Wanglin

Uji Ke-	Waktu Komputasi (ms)
1	16832
2	15887
3	16035
4	16531
5	16247

6	16274
7	16140
8	17013
9	15908
10	16313
Total	163180 ms
Rata-rata	16318 ms

Dari 10 kali percobaan yang dilakukan terhadap 10 apel wanglin yang berbeda, didapat rata-rata waktu komputasi sebesar 16318 ms atau sama dengan 16 detik

Tabel 6.10 Hasil total waktu komputasi keempat apel

No	Nama	Rata-rata waktu komputasi
1	Apel ana	15713.5 ms
2	Apel manalagi	15980.5 ms
3	Apel romebeauty	15976.8 ms
4	Apel wanglin	16318 ms
Total		63988.8 ms
Rata-rata		15997.2 ms

Berdasarkan hasil pengujian diatas terdapat 40 kali pengujian yang telah dilakukan, dan mendapat total waktu sebesar 63988.8 ms dan rata-rata yaitu 15997.2 ms atau sama dengan 15 detik. Berikut akan ditampilkan salah satu *output* dari percobaan pertama untuk jenis apel ana.



```
Apel Ana
Waktu Komputasi : 15856 ms
```

Gambar 6.2 Output Waktu Komputasi

6.3 Pengujian Akurasi Klasifikasi Dari Sisi *Software* Dan *Hardware*

6.3.1 Tujuan Pengujian

Pengujian ini dilakukan bertujuan untuk melihat apakah hasil *output* dari compiler program dengan aktuator berjalan sesuai dengan kondisi yang sebenarnya.

6.3.2 Prosedur Pengujian

Prosedur pengujian dilakukan agar pengujian berjalan dengan semestinya, adapun langkah-langkah yang dilakukan adalah sebagai berikut :

1. Menghubungkan *powersupply* ke sumber tegangan untuk menyalakan seluruh komponen.

2. Upload program ke Arduino Uno.
3. Menjalankan program pada opencv untuk pengolahan citra.
4. Meletakkan objek pada konveyor pada jarak yang sudah ditentukan dari posisi webcam.
5. Hasil berupa nilai *Hue*, *Aspectratio* dan *GLCM Contrast* akan muncul pada *compiler* program.
6. Melihat apakah servo akan mendorong buah apel sesuai dengan jenisnya.

6.3.3 Pelaksanaan Pengujian

Pelaksanaan pengujian dilakukan dengan mendeteksi ke 40 jenis apel dan melihat apakah servo bergerak sesuai dengan jenis apelnya.

6.3.4 Hasil dan Analisis Pengujian

Berikut adalah hasil servo mendorong buah apel.

Tabel 6.11 Akurasi *Hardware* dan *Software* saat mendorong Apel Ana

Uji Ke-	Hasil Klasifikasi	Apel Sebenarnya	Kesesuaian	Servo Mendorong
1	Ana	Ana	Sesuai	Ya
2	Ana	Ana	Sesuai	Ya
3	Ana	Ana	Sesuai	Ya
4	Ana	Ana	Sesuai	Ya
5	Ana	Ana	Sesuai	Ya
6	Ana	Ana	Sesuai	Ya
7	Ana	Ana	Sesuai	Ya
8	Ana	Ana	Sesuai	Ya
9	Ana	Ana	Sesuai	Ya
10	Tidak Terdeteksi	Ana	Tidak Sesuai	Tidak
Hasil Akurasi				90%

Berdasarkan hasil pengujian yang sudah dilakukan untuk buah apel ana, jika parameter nilai tidak terpenuhi maka *compiler* tidak akan menampilkan hasil sehingga servo tidak akan melakukan gerakan apapun.

Untuk menghitung akurasi kesesuaian *Hardware* dan *Software* saat mendorong Apel Ana adalah sebagai berikut :

$$Akurasi = \frac{Total\ data - Data\ tidak\ sesuai}{Total\ Data} \times 100\%$$

$$\begin{aligned}
 Akurasi &= \frac{10 - 1}{10} \times 100\% \\
 &= 90\%
 \end{aligned}$$

Tabel 6.12 Akurasi *Hardware* dan *Software* saat mendorong Apel Manalagi

Uji Ke-	Hasil Klasifikasi	Apel Sebenarnya	Kesesuaian	Servo Mendorong
1	Manalagi	Manalagi	Sesuai	Ya
2	Manalagi	Manalagi	Sesuai	Ya
3	Manalagi	Manalagi	Sesuai	Ya
4	Manalagi	Manalagi	Sesuai	Ya
5	Manalagi	Manalagi	Sesuai	Ya
6	Manalagi	Manalagi	Sesuai	Ya
7	Manalagi	Manalagi	Sesuai	Ya
8	Manalagi	Manalagi	Sesuai	Ya
9	Manalagi	Manalagi	Sesuai	Ya
10	Tidak Terdeteksi	Manalagi	Tidak Sesuai	Tidak
Hasil Akurasi				90%

Berdasarkan hasil pengujian yang sudah dilakukan untuk buah apel manalagi, jika parameter nilai tidak terpenuhi maka *compiler* tidak akan menampilkan hasil sehingga servo tidak akan melakukan gerakan apapun.

Untuk menghitung akurasi kesesuaian *Hardware* dan *Software* saat mendorong Apel Manalagi adalah sebagai berikut :

$$Akurasi = \frac{Total\ data - Data\ tidak\ sesuai}{Total\ Data} \times 100\%$$

$$Akurasi = \frac{10 - 1}{10} \times 100\%$$

$$= 90\%$$

Tabel 6.13 Akurasi *Hardware* dan *Software* saat mendorong Apel Romebeauty

Uji Ke-	Hasil Klasifikasi	Apel Sebenarnya	Kesesuaian	Servo Mendorong
1	Romebeauty	Romebeauty	Sesuai	Ya
2	Romebeauty	Romebeauty	Sesuai	Ya
3	Romebeauty	Romebeauty	Sesuai	Ya
4	Romebeauty	Romebeauty	Sesuai	Ya
5	Romebeauty	Romebeauty	Sesuai	Ya
6	Romebeauty	Romebeauty	Sesuai	Ya
7	Tidak Terdeteksi	Romebeauty	Tidak Sesuai	Tidak
8	Tidak Terdeteksi	Romebeauty	Tidak Sesuai	Tidak
9	Tidak Terdeteksi	Romebeauty	Tidak Sesuai	Tidak
10	Tidak Terdeteksi	Romebeauty	Tidak Sesuai	Tidak
Hasil Akurasi				60%

Berdasarkan hasil pengujian yang sudah dilakukan untuk buah apel romebeauty, jika parameter nilai tidak terpenuhi maka *compiler* tidak akan menampilkan hasil sehingga servo tidak akan melakukan gerakan apapun.

Untuk menghitung akurasi kesesuaian *Hardware* dan *Software* saat mendorong Apel Romebeauty adalah sebagai berikut :

$$Akurasi = \frac{Total\ data - Data\ tidak\ sesuai}{Total\ Data} \times 100\%$$

$$Akurasi = \frac{10 - 4}{10} \times 100\%$$

$$= 60 \%$$

Tabel 6.14 Akurasi *Hardware* dan *Software* saat mendorong Apel Wanglin

Uji Ke-	Hasil Klasifikasi	Apel Sebenarnya	Kesesuaian	Servo Mendorong
1	Wanglin	Wanglin	Sesuai	Ya
2	Wanglin	Wanglin	Sesuai	Ya
3	Wanglin	Wanglin	Sesuai	Ya
4	Wanglin	Wanglin	Sesuai	Ya
5	Wanglin	Wanglin	Sesuai	Ya
6	Wanglin	Wanglin	Sesuai	Ya
7	Wanglin	Wanglin	Sesuai	Ya
8	Wanglin	Wanglin	Sesuai	Ya
9	Tidak Terdeteksi	Wanglin	Tidak Sesuai	Tidak
10	Tidak Terdeteksi	Wanglin	Tidak Sesuai	Tidak
Hasil Akurasi				80%

Berdasarkan hasil pengujian yang sudah dilakukan untuk buah apel wanglin, jika parameter nilai tidak terpenuhi maka *compiler* tidak akan menampilkan hasil sehingga servo tidak akan melakukan gerakan apapun.

Untuk menghitung akurasi kesesuaian *Hardware* dan *Software* saat mendorong Apel Wanglin adalah sebagai berikut :

$$Akurasi = \frac{Total\ data - Data\ tidak\ sesuai}{Total\ Data} \times 100\%$$

$$Akurasi = \frac{10 - 2}{10} \times 100\%$$

$$= 80 \%$$

Dari hasil pengujian ke 40 jenis apel diatas servo akan bergerak jika apel terdeteksi sesuai dengan jenisnya. Maka dapat rata-rata akurasi dari software dan hardware adalah sebagai berikut :

Tabel 6.15 Rata-rata akurasi kesesuaian *Hardware* dan *Software*

No	Nama	Rata-rata <i>hardware</i> dan <i>software</i>
1	Apel ana	90%
2	Apel manalagi	90%

3	Apel romebeauty	60%
4	Apel wanglin	80%
Rata-rata		80%

Dari Tabel 6.15 didapat hasil akurasi dari pengujian keempat jenis tersebut sebesar 80%.



BAB 7 PENUTUP

7.1 Kesimpulan

Dari percobaan yang telah dilakukan oleh penulis, dapat ditarik kesimpulan sebagai berikut :

1. Mendeteksi nilai *aspectratio* keempat jenis apel dengan menggunakan *webcam* didapat hasil masing-masing buah adalah buah Ana 90%, Manalagi 90%, Romebeauty 60% dan Wanglin sebesar 80%. Sedangkan untuk akurasi keseluruhan adalah sebesar 80%. Hasil akurasi ini dinilai baik, memang terdapat perbedaan ukuran dari perhitungan manual dengan sistem. Perhitungan manual yang menggunakan penggaris dan bentuk apel yang tidak datar ini memungkinkan terjadi perbedaan angka yang kecil yang tidak bisa dihitung dengan menggunakan penggaris pada umumnya. Sehingga presentase error untuk buah apel ana sebesar 21.5 % sedangkan untuk buah apel manalagi sebesar 7.369 %.
2. Sistem ini dapat menampilkan waktu komputasi dari masing-masing buah apel yang dideteksi. Waktu komputasi yang dibutuhkan sistem untuk mendeteksi sampai proses klasifikasi membutuhkan waktu total rata-rata 15997.2 ms yang jika dikonversi ke detik menjadi 15 detik. Waktu yang lama ini dinilai wajar dikarenakan beban komputasi yang dilakukan untuk mendeteksi tekstur apel ini berat.
3. Setelah dilakukan pengujian akurasi antara *hardware* dan *software* didapat nilai akurasi untung masing-masing buah adalah buah Ana 90%, Manalagi 90%, Romebeauty 60% dan Wanglin sebesar 80%. Sedangkan untuk akurasi keseluruhan adalah sebesar 80%. Terdapat 8 jenis apel yang tidak terdeteksi oleh *software*, ini dikarenakan kulit buah apel yang tidak terdeteksi memiliki nilai diluar dari range yang sudah ditentukan. Oleh karena itu servo tidak bergerak untuk melakukan proses klasifikasi.

7.2 Saran

Berikut saran-saran yang dapat digunakan sebagai acuan pengembangan penelitian ini ataupun yang serupa kedepannya:

1. Menambahkan metode lain untuk mendeteksi tekstur kulit agar hasil yang diinginkan sesuai dengan keadaan apel sesungguhnya.
2. Menggunakan dua webcam dari kedua sisi untuk memaksimalkan pendeteksian objek.
3. Menambah satu buah lengan pendorong untuk klasifikasi apel wanglin.

DAFTAR PUSTAKA

- AL Amin Putra, K. I. H., 2013. Perancangan Dan Pembuatan Alat Pemisah Buah Apel Berdasarkan Ukuran Dengan Pengendali Mikrokontroler Atmega 8535. pp. 2-3.
- Amrullah, S. A., 2017. Perancangan Sistem Inspeksi Visual Berbasis Computer Vision Untuk Penggolongan Buah Apel. p. 9.
- Anton Yudhana, S. S. S., 2016. Perbandingan Segmentasi Pada Citra Asli dan Citra Kompresi Wavalet untuk Identifikasi Telur. Volume 8, p. 3.
- Auliia, A., 2013. *Jenis-Jenis Apel*. [Online]
Available at: <http://thecrazy-apple.blogspot.com/2013/03/jenis-jenis-apel.html>
[Accessed 15 August 2018].
- Fajar Ridho Wicaksono, A. R. I. D. W., 2018. Perancangan Dan Implementasi Alat Penyortir Barang Pada Konveyor Dengan Pengolahan Citra. Volume 5, pp. 5-6.
- Fauzan, A., 2016. *Mengkonversi Citra Berwarna, Citra Keabuan dan Citra Biner*. [Online]
Available at: <http://www.charisfauzan.net/2016/02/bagaimana-mengkonversi-citra-berwarna.html>
[Accessed 16 August 2018].
- Fitriyah, H., 2016. Modul Ajar Komputasi Citra dan Suara Digital. pp. 4-7.
- Habibi, M., 2016. *Perbedaan Motor Stepper dan Motor Servo*. [Online]
Available at: <https://muhammadhabibieblog.wordpress.com/2016/05/17/perbedaan-motor-stepper-dan-motor-servo/>
[Accessed 14 March 2018].
- Hebert Adrianto, D. H. D. T. S., 2013. Biosistematika Varietas Pada Apel (*Malus Sylvestris L.*) Di Kota Batu Berdasarkan Morfologi. p. 8.
- Kurniati, F. T., 2015. Pengembangan Printer Forensik Untuk Identifikasi Data Dokumen Cetak. p. 14.
- Logitech, 2018. *HD WEBCAM C310*. [Online]
Available at: <https://www.logitech.com/id-id/product/hd-webcam-c310>
[Accessed 22 Maret 2018].
- MATT, 2016. *Introducing The Raspberry Pi 3 Model B Single Board Computer*. [Online]
Available at: <https://www.raspberrypi-spy.co.uk/2016/02/introducing-the-raspberry-pi-3-model-b/>
[Accessed 21 March 2018].

- Mohammad Fauzin Amin, S. R. A. E. R. W., 2017. Rancang Bangun Sistem Sortir Buah Apel Menggunakan Sensor Warna Dan Sensor Suhu. Volume III, p. 2.
- Monda, 2015. *Belajar Foto di Wisata Petik Apel Malang*. [Online] Available at: <https://www.mondasiregar.com/belajar-foto-di-wisata-petik-apel-malang/> [Accessed 15 8 2018].
- Nafidha, P. N., 2016. Aplikasi Pendeteksi Objek Bergerak pada Image Sequence dengan Metode Background Substraction. Volume 21, p. 8.
- Nedelkovski, D., 2018. *How To Control a Stepper Motor with A4988 Driver and Arduino*. [Online] Available at: <https://howtomechatronics.com/tutorials/arduino/how-to-control-stepper-motor-with-a4988-driver-and-arduino/> [Accessed 1 July 2017].
- Neneng, Y. F., 2017. Klasifikasi Jenis Daging Berdasarkan Analisis Citra Tekstur Gray Level Co-Occurrence Matrices (GlcM) Dan Warna. pp. 2-3.
- Nizam, A. S., 2015. *Aspek Rasio dalam Dunia Multimedia*. [Online] Available at: <http://ipulgates.blogspot.com/2015/01/aspek-rasio-dalam-dunia-multimedia.html> [Accessed 17 August 2018].
- Pamungkas, A., 2016. *Pemrograman Matlab*. [Online] Available at: <https://pemrogramanmatlab.com/2016/11/09/segmentasi-warna-citra-digital/> [Accessed 16 August 2018].
- Paulhurleyuk, 2018. *Instructables*. [Online] Available at: <https://www.instructables.com/id/Drive-a-Stepper-Motor-with-an-Arduino-and-a-A4988-/> [Accessed 15 August 2018].
- Riadi, M., 2016. *Pengolahan Citra Digital*. [Online] Available at: <https://www.kajianpustaka.com/2016/04/pengolahan-citra-digital.html> [Accessed 1 June 2018].
- RD. Kusumanto, W. S. P. A. N. T., 2012. *Aplikasi Sensor Vision untuk Deteksi MultiFace dan Menghitung Jumlah Orang*. Palembang, Polteknik Negeri Sriwijaya.
- Saifudin, A. F., 2015. Sistem Identifikasi Citra Kayu Berdasarkan Tekstur Menggunakan Gray Level Coocurrence Matrix (GlcM) Dengan Klasifikasi Jarak Euclidean. Volume 19, pp. 2-5.
- Satrio Sani Sadewo, R. S. I. C., 2015. Sistem Pengukur Kecepatan Kendaraan Berbasis Pengolahan Video. *IJEIS*, Volume 5, p. 5.

- Sutopo, 2015. Balai Penelitian Tanaman Jeruk dan Buah Subtropika BALITBANGTAN - Kementerian Pertanian. [Online]
Available at: <http://balitjestro.litbang.pertanian.go.id/budidaya-apel/>
[Accessed 19 September 2018].
- Supply, P., 2018. *Pi Supply*. [Online]
Available at: <https://uk.pi-supply.com/products/raspberry-pi-camera-board-v1-3-5mp-1080p>
[Accessed 21 March 2018].
- Tri Arya Nugraha, S. S. I. H., 2017. *Aplikasi Pendeteksian Tangan dalam Ruang Warna YCbCr dan HSV pada Virtual Mouse*. Palembang, Politeknik Negeri Sriwijaya, p. 4.
- Venantius, 2013. *Simple Aja*. [Online]
Available at: <https://simple-aja.info/aspect-ratio-pada-layar-televisi-dan-video.html>
[Accessed 17 August 2018].
- Wisata, C., 2015. *Outbound Dan Petik Buah Di Kusuma Agrowisata Batu*. [Online]
Available at: <https://www.calyawisata.com/2015/04/outbound-dan-petik-buah-di-kusuma-agrowisata.html>
[Accessed 16 August 2018].
- Yosefina, A., 2015. *Kandungan Buah Apel Dan Manfaatnya Bagi Kesehatan*. [Online]
Available at: <http://www.kiat-sehat.com/kandungan-buah-apel-dan-manfaatnya-bagi-kesehatan/>
[Accessed 9 February 2018].
- Yudhistira Ganis K, I. S. R. R. I., 2015. Klasifikasi Citra Dengan Matriks Ko-Okurensi Aras Keabuan (Gray Level Co-Occurrence Matrix-Glcm) Pada Lima Kelas Biji-Bijian. p. 3.
- Yuwono, S. S., 2015. *Apel (Malus sylvestris Mill)*. [Online]
Available at: <http://darsatop.lecture.ub.ac.id/2015/09/apel-malus-sylvestris-mill/>